

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Comunicações

Análise de Uso de Sociedade de Tutores Inteligentes com Aplicação em Sistemas de e-Gov

Autor: Ekler Paulino de Mattos
Orientador: Leonardo de Souza Mendes

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação (FEEC–UNICAMP) como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Telecomunicações e Telemática. Aprovação em: 07/08/2007.

Banca Examinadora:

Prof. Dr. Maurício Ferreira Magalhães
Prof. Dr. Max H. M. Costa
Prof. Dr. Mario Lemes Proença Jr.

UNICAMP
UNICAMP
UEL

Campinas, SP
Agosto/2007

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

M436a Mattos, Ekler Paulino de
 Análise de uso de sociedade de tutores inteligentes com
 aplicação em sistemas de e-Gov / Ekler Paulino de Mattos. -
 -Campinas, SP: [s.n.], 2007.

 Orientador: Leonardo de Souza Mendes
 Dissertação (Mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação.

 1. Inteligência artificial. 2. Inteligência artificial
distribuída. 3. Administração de material. I. Mendes,
Leonardo de Souza. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Título em Inglês: Analysis of Intelligent Tutors Society in e-Gov Systems

Palavras-chave em Inglês: Intelligent tutorial systems, Cognitive multiagent systems,
Management of materials, Distributed artificial intelligence

Área de concentração: Telecomunicações e Telemática

Titulação: Mestre em Engenharia Elétrica

Banca examinadora: Maurício Ferreira Magalhães, Max Henrique Machado Costa, Mario
Lemes Proença Junior

Data da defesa: 07/08/2007

Programa de Pós-Graduação: Engenharia Elétrica

COMISSÃO JULGADORA – TESE DE MESTRADO

Candidato(a): Ekler Paulino de Mattos

Data da Defesa: 07 de Agosto de 2007

Título da Tese: “Análise de Uso de Sociedade de Tutores Inteligentes com Aplicação em Sistemas de e-Gov”

Prof. Dr. Leonardo de Souza Mendes (Presidente):

Prof. Dr. Mario Lemes Proença Junior:

Prof. Dr. Max Henrique Machado Costa:

Prof. Dr. Mauricio Ferreira Magalhães:

Resumo

O *Sistema Tutor Inteligente* (STI) pertence a uma categoria de sistemas de natureza educacional, utilizado como ferramenta de suporte ao ensino-aprendizagem. Possui uma estrutura modular que tem por finalidade auxiliar o aprendiz na realização de atividades educacionais, bem como a capacidade de adaptar-se de acordo com as necessidades de um aprendiz, o que faz do STI uma arquitetura interessante na construção de softwares educacionais.

O trabalho proposto tem por objetivo utilizar a arquitetura de STI, aplicada à área de sistemas de e-Gov como proposta de solução de problemas de natureza distribuída. Como estudo de caso, foi escolhida a área Gestão de Materiais e Medicamentos, justamente por apresentar problema pertinente à distribuição de materiais e medicamentos, nas unidades básicas de saúde (UBS). Cada STI funciona como representante de uma UBS, que tem por função realizar o papel de um agente gestor de estoque (*Agente Gestor Tutor - AGT*), cargo pouco comum na rede municipal de saúde, auxiliando o administrador de cada setor (visto como o aprendiz) a realizar tarefas complexas de gestão de materiais e medicamentos. Foi realizada uma série de simulações usando o protótipo desenvolvido para testar a sua viabilidade de aplicação com relação ao tratamento do estoque distribuído de uma arquitetura de rede municipal de saúde.

Palavras-chave: Sistemas Tutores Inteligentes, Sistemas Multi-Agentes Cognitivos, Gestão de Materiais, Inteligência Artificial Distribuída.

Abstract

The Intelligent Tutor Systems (ITS) belong to a category of educational nature systems, used as a tool to support the teaching and learning. It has a modular structure, which aim to help the apprentice in the execution of educational activities, as well as in adapting itself according to the apprentice's necessities, what makes the ITS, an interesting architecture in the construction of educational softwares.

The proposed work aim to use the ITS architecture in the management of materials, as a solution for the problem of medicine distribution in the health basic units (HBU). Each ITS works as a HBU representative, whose function is to play the role of a managing agent of supply (Tutorial Managing Agent - TMA), a post job not so common in the municipal health's network. The TMA assists the administrator of each sector (seen as the apprentice) in executing complex tasks of management of materials and medicines. In this way, many simulations were carried out, using the developed prototype to test its feasibility of application, in relation to the management of the distributed materials of architecture of a municipal health's network.

Key-Words: Intelligent Tutorial Systems, Cognitive Multiagent Systems, Management of Materials, Distributed Artificial Intelligence.

Agradecimentos

Inicialmente, gostaria de agradecer ao nosso pai amado, arquiteto do universo. Ao meu Deus, por ter concebido a possibilidade de estar aqui neste momento. Agradecer por ter conseguido algo que tantos anseiam neste país, cursar uma faculdade e concluir uma pós-graduação de qualidade.

À Nossa Senhora da Aparecida, minha mãe amada, que me afagou nas noites em que senti sozinho.

Aos meus pais, Mattos e Elisa, por serem a base de tudo o que sou, meus anjos da guarda que estiveram ao meu lado em todos os capítulos de minha vida.

À minha irmã Thaiz, pessoa que transformou os momentos de tristeza em boas piadas, lágrimas no mais belo sorriso. Um exemplo de mulher distinta e batalhadora. À minha irmã Rosane e sua família: Silvio, Amanda e Leonardo, as minhas mais novas jóias.

Aos meus grandes primos(as) e amigos(as): Leandro, Marcos Vinícius, Criseverton, Andréa, Maria Paula, Ana Carolina e também a todos os familiares que sempre me apoiaram nas minhas escolhas e que se fizeram presentes, principalmente nos momentos de dificuldade.

Ao meu amor, meu alicerce, ser iluminado em que sempre me espelho para me tornar uma pessoa melhor. Pessoa com a qual dividi todos os meus momentos, desde o primeiro dia em que a conheci. Obrigado pelo apoio técnico para construir esta obra e por fazer parte de minha vida, Ana Carolina.

À minha segunda família, meus primos Elisa e Hugo, pessoas que me acolheram como se fosse um filho em Campinas. Aos seus filhos, meus primos: Sarah Sophia, Victor Hugo, Isabelle Elisa e Arthur Henrique, que fizeram de meus dias a mais doce brincadeira-de-criança. Também não poderia de deixar de agradecer à Fabiana, minha prima, que sonhou em ser mestre e conseguiu.

Aos amigos Maurício Bottoli, Gean Breda e Cláudio Araújo, pelo apoio técnico e pelas inúmeras considerações feitas no decorrer de minha caminhada. Pessoas que, definitivamente, foram o meu norte para atingir minha meta.

A todos os companheiros do LarCom (Laboratório de Comunicações e Redes de Computadores).

Ao colega Rodrigo Rossato e também à Prefeitura Municipal de Campinas, pelo fornecimento dos dados dos setores públicos, informações imprescindíveis para a obtenção dos resultados de meu trabalho.

Aos meus amigos de infância e adolescência que, de alguma forma, contribuíram para a produção desta dissertação através de suas motivações e sonhos no passado. Ailson Zanchett, Paulo Lindomar, Francimar Oliveira, Márcio Nunes, Robinson Müller, Izaias Melo, Athos Messias, Américo Nogueira, Fernando Paro e Eduardo Paro.

Aos amigos que praticam Ninjutsu (Arte Ninja): Celso, Evandro, Liane, Leandro, Guilherme, Fábio e meu shidoshi Fred, que através de seus ensinamentos me levou ao encontro da serenidade nos momentos de tensão.

Meus agradecimentos também vão às meninas: Letícia, garota que sou fã número 1; Ana Carolina, meu amor; Karina, grande amiga; Fernanda, Letícia e Daniela, todas integrantes da república da Casa da Mãe Joana e ao agregado Miguel, grande amigo e companheiro.

Aos meus amigos e confidentes Ricardo Alberti, Marcelo Cubas e Bruno Zarpelão, pelos anos de convivência e imensurável apoio.

Ao professor e notável amigo Odival Faccenda, a principal fonte de inspiração para seguir a estrada da vida acadêmica.

À pessoa que literalmente abriu a porta de sua sala, me acolheu com todo carinho, apoiou e apostou em meu potencial. Gostaria de agradecer ao orientador Leonardo de Souza Mendes, pela orientação, apoio e companheirismo no decorrer destes ótimos anos.

Dedico aos meus pais: Mattos e
Elisa; e irmãs: Thaiz e Rosane.

In memoriam:
Eurico Gonçalves de Mattos,
Alfredo Bossoni e
João Navachi.

Sumário

Lista de Figuras.....	xiv
Lista de Tabelas	xvii
Glossário	xix
1 Introdução	1
1.1 Motivação.....	2
1.2 Objetivo.....	3
1.3 Disposição dos Capítulos	4
2 Sistema Tutores Inteligentes e Agentes	5
2.1 Introdução	5
2.2 Estado da Arte dos STI	5
2.3 Arquitetura de um STI	6
2.3.1 Módulo do Aprendiz	7
2.3.2 Módulo Tutor	9
2.3.3 Módulo de Domínio	10
2.3.4 Módulo Interface.....	11
2.4 STI e Sistemas Multi-Agentes	11
2.4.1 Inteligência Artificial Distribuída (IAD)	12
2.4.2 Agentes Inteligentes e sua Definição	12
2.4.3 Arquiteturas de Agentes.....	15
2.5 A Linguagem AgentSpeak(L).....	17
2.6 Sistemas Multi-Agentes (SMA).....	19
2.6.1 A evolução dos SMA	21
2.6.2 Tipos de SMA	21
2.7 A Linguagem de Comunicação entre Agentes KQML e o Protocolo de Interação Mecanismo de Contrato Orientado a Agentes Facilitadores	22
2.7.1 Linguagem de comunicação entre agentes (ACL).....	22
2.7.2 A Linguagem de Comunicação entre Agentes KQML.....	23
2.8 O Protocolo de Interação entre Agentes – Mecanismo de Contrato Orientado a Agentes Facilitadores	28
2.9 Bibliografia	31
3 O Sistema de Gestão de Materiais e Medicamentos - Sig2m	35
3.1 Histórico.....	35
3.2 A Arquitetura da Rede Municipal de Saúde – Distribuição de Materiais.....	35
3.2.1 Disposição dos Setores	35
3.2.2 Disposição de Hardware e Software	36
3.2.3 Tecnologias Utilizadas para a Construção do Sig2m.....	37
3.2.4 Arquitetura Sistêmica.....	38
3.3 Principais Limitações do Sistema	40
3.3.1 Arquitetura cliente-servidor	40
3.3.2 Módulo de Planejamento de Compra e Gestão de Estoque	41
3.3.3 Facilidade de Uso.....	41
3.4 Soluções propostas.....	42

3.4.1	Limitações Arquiteturais.....	42
3.4.2	Soluções propostas para as Limitações de Gestão de Materiais e Recursos Humanos ..	49
3.5	Bibliografia	51
4	Abordagem de Sociedades de STI como instrumento de auxílio à gestão de materiais e medicamentos.....	53
4.1	Introdução	53
4.2	Arquitetura da Rede Municipal de Saúde	53
4.2.1	O processo de distribuição de materiais.....	53
4.3	Modelagem das sociedades (Solução Proposta)	55
4.3.1	Sociedade Local	55
4.3.2	Sociedade Remota.....	56
4.3.3	Arquitetura das sociedades.....	58
4.3.4	Protocolo de comunicação entre agentes	59
4.4	Funcionamento do Sistema	59
4.4.1	Trabalho cooperativo da Sociedade Remota.....	62
4.5	Arquitetura implementada (Componentes do sistema).....	62
4.5.1	Modelagem do Agente Gestor Tutor (AGT).....	63
4.5.2	Modelagem matemática do saldo ideal para um estoque.....	77
4.6	Componentes da Arquitetura Clássica de um STI sobre Arquitetura Implementada	92
4.6.1	Módulo de Domínio	93
4.6.2	Módulo do Tutor	93
4.6.3	Interface	94
4.6.4	Módulo do Aprendiz	95
4.7	Síntese do Capítulo	99
4.8	Bibliografia	100
5	Resultados Obtidos	103
5.1	Introdução	103
5.2	Estrutura da Simulação	103
5.3	Síntese do processo de distribuição de materiais e medicamentos para os setores da rede municipal de saúde.....	105
5.4	Notas sobre a simulação.....	106
5.5	Resultados da coreografia do histórico do estoque real	106
5.5.1	Análise dos Resultados do Material 1 nas Unidades Básicas de Saúde.....	106
5.5.2	Análise dos Resultados do Material 2 nas Unidades Básicas de Saúde.....	111
5.5.3	Análise dos Resultados do Material 3 nas Unidades Básicas de Saúde.....	116
5.6	Conclusão da Análise do Estoque real e Ideal	120
5.7	Análise da coreografia dos Parâmetros do Estado Geral do Estoque (EGE).....	120
5.7.1	Introdução	120
5.7.2	Análise dos parâmetros do EGE da UBS 1	121
5.7.3	Análise dos parâmetros do EGE da UBS 2.....	122
5.7.4	Análise dos parâmetros do EGE da UBS 3.....	124
5.7.5	Análise dos parâmetros do EGE da UBS 4.....	125
5.7.6	Análise dos parâmetros do EGE da UBS 5.....	126
5.8	Considerações finais sobre o EGE perante a sociedade de AGTs	128
6	Discussão e Conclusões	129
6.1	Contribuições	129
6.2	Trabalhos Futuros	132
6.3	Bibliografia	133

Anexo A – Histórico do Estoque dos Materiais das UBSs	135
A.1. Histórico do Estoque dos Materiais 4 e 5 da UBS 1	135
A.2. Histórico do Estoque dos Materiais 4 e 5 da UBS 2	136
A.3. Histórico do Estoque dos Materiais 4 e 5 da UBS 3	137
A.4. Histórico do estoque dos materiais 4 e 5 da UBS 4	138
A.5. Histórico do estoque dos materiais 4 e 5 da UBS 5	139
Anexo B - Gestão de materiais e medicamentos.....	141
B.1. Introdução	141
B.2. Gestão de Estoque Função e Atividades	141
B.2.1. Fundamentos	141
B.2.2. Almoxarifado	142
B.3. Componentes da administração de materiais	143
B.3.1. Estoque.....	143
B.3.2. Consumo	143
B.3.3. Demanda	144
B.3.4. Previsões de demanda	144
B.4. Classificação de materiais.....	149
B.4.1. Tipos de classificação	149
B.5. Modelagem matemática da gestão de materiais: O ressuprimento	153
B.5.1. Estoque Máximo (EMx) e Estoque Virtual (EV).....	154
B.5.2. Estoque Mínimo ou de Segurança	155
B.5.3. Tempo de Reposição.....	156
B.5.4. Quando Comprar (QC)	157
B.5.5. Ponto de Pedido (PP)	158
B.5.6. Quanto Comprar.....	158
B.6. Síntese do capítulo	160
B.7. Bibliografia	160
Anexo C - Modelagem UML das ações e percepções dos elementos da sociedade de Tutores Inteligentes.....	163
C.1. Diagrama de Classes dos Principais processos de ação do AGT.....	163
C.1.1. Diagrama de classe Calcular a Lista de Materiais que irão Exceder e Zerar.....	163
C.1.2. Diagrama de classe Calcular o Estado Geral do Estoque	165
C.1.3. Diagramas de classe de entrar na sociedade local e remota.....	167
C.2. Diagramas de subcasos de uso e diagramas de classe das ações do AGT	167
C.3. Diagrama de Classes dos Principais processos de percepção do AGT.....	175
Anexo D – Mensagens KQML do Protocolo de Negociação de Requisição e Doação de Medicamentos, realizado pela Sociedade Remota	177
D.1. Mensagens KQML enviadas pelos agentes comunicativos de cada UBS	178
Anexo E – Artigo apresentado no CBIS - X Congresso Brasileiro de Informática em Saúde "Informática em Saúde e Cidadania". Florianópolis – SC.....	183

Lista de Figuras

Figura 1 - Arquitetura clássica de um STI.....	7
Figura 2 - Modelo geral de agente proposto por [WOO1999].....	15
Figura 3 - Arquitetura BDI Genérica (adaptado de por [WOO1999]).....	16
Figura 4 - Código AgentSpeak(L) – estrutura de um plano.....	18
Figura 5 - Exemplo de código AgentSpeak(L), adaptado de [HUB2004].....	19
Figura 6 – Estrutura de uma mensagem de um agente.	23
Figura 7 - Estrutura básica de uma mensagem KQML, extraído de [HUB2003].....	24
Figura 8 - Exemplo de uso do protocolo KQML.....	26
Figura 9 - Protocolo de interação de rede de contrato.	30
Figura 10 - Sociedade de agentes intermediada pelo agente facilitador.....	31
Figura 11 - A arquitetura da rede municipal de saúde.....	36
Figura 12 - Estrutura da rede municipal de saúde – visão de software.....	37
Figura 13 - Arquitetura cliente – servidor.....	39
Figura 14 - Modelo de passagem de mensagens - Sockets.....	40
Figura 15 - Arquitetura 3 – Tier, (Extraído de [CAR2007]).....	43
Figura 16 - Arquitetura MVC – <i>Model, View e Controller</i>	45
Figura 17 - Implementação do Sig2m sob arquitetura 3 -Tier.....	46
Figura 18 - Funcionamento da arquitetura de web services (Extraído de [TID2001]).....	47
Figura 19 - Arquitetura de SOA - Web Services, adaptado de [TID2001].....	48
Figura 20 - Implementação do Sig2m sob arquitetura de Web Services.....	49
Figura 21 - Arquitetura da rede de Municipal de Saúde (UBSs e Almoxarifado Gestor).	54
Figura 22 - Arquitetura da sociedade local.....	56
Figura 23 -Disposição da Sociedade Remota sobre a Rede Municipal de Saúde.....	57
Figura 24 - Processo de requisição e doação de materiais entre UBSs.....	58
Figura 25 - Sociedade intermediada pelo agente facilitador.....	59
Figura 26 - Ciclo de tarefas do AGT.....	60
Figura 27 – Detalhes do processo de requisição e doação de medicamentos entre UBSs.....	61
Figura 28 - Funcionamento do Interpretador Jason.	64
Figura 29 - Anatomia do Agente Gestor Tutor.....	65
Figura 30 - Classe Java Agente Gestor Tutor – TutorManagerAgent.	66
Figura 31 - Diagrama de Classe apresenta a ligação entre a classe TutorManagerAgent e o interpretador Jason.....	67
Figura 32 - Caso de uso que descreve as principais percepções realizadas pelo AGT.....	69
Figura 33 - Diagrama de classes das percepções do AGT.....	70
Figura 34 - Caso de Uso das ações do AGT.....	74
Figura 35 - Diagrama de classes das ações do AGT.....	75
Figura 36– Situação ideal do estoque.	78
Figura 37 – Constituição do núcleo do AGT.....	79
Figura 38 - Estrutura de planos a camada de gestão do estoque.....	81
Figura 39 - Estado Geral do Estoque.....	82
Figura 40 - Estrutura de planos para o processo de requisição e doação para a sociedade remota.	86
Figura 41 - Diagrama de classe de envio e recebimento de mensagens locais e remotas do AGT.	88
Figura 42 - Arquitetura do Agente Mensageiro.....	89

Figura 43 - Diagrama de caso de uso do Agente Mensageiro.	90
Figura 44 - Diagrama de classes do AM.	91
Figura 45 - Disposição dos Processos AGT, AM e Sig2m.	92
Figura 46 - Disposição dos componentes da arquitetura do STI sobre a arquitetura implementada.	93
Figura 47 - Ciclo de distribuição de materiais e medicamentos na rede municipal de saúde.	105
Figura 48 - histórico do estoque do material 1, UBS 1.	107
Figura 49- histórico do estoque do material 1, UBS 2.	108
Figura 50- histórico do estoque do material 1, UBS 3.	109
Figura 51 - histórico do estoque do material 1, UBS 4.	110
Figura 52 - histórico do estoque do material 1, UBS 5.	111
Figura 53 - histórico do estoque do material 2, UBS 1.	112
Figura 54 - histórico do estoque do material 2, UBS 2.	113
Figura 55 - histórico do estoque do material 2, UBS 3.	114
Figura 56 - histórico do estoque do material 2, UBS 4.	114
Figura 57 - histórico do estoque do material 2, UBS 5.	115
Figura 58 - histórico do estoque do material 3, UBS 1.	116
Figura 59 - histórico do estoque do material 3, UBS 2.	117
Figura 60 - histórico do estoque do material 3, UBS 3.	118
Figura 61 - histórico do estoque do material 3, UBS 4.	118
Figura 62 - histórico do estoque do material 3, UBS 5.	119
Figura 63 - Estado geral do estoque da UBS 1.	121
Figura 64 - Estado geral do estoque da UBS 2.	123
Figura 65 - Estado geral do estoque da UBS 3.	124
Figura 66 - Estado geral do estoque da UBS 4.	126
Figura 67 - Estado geral do estoque da UBS 5.	127
Figura 68 - histórico do estoque do material 4, UBS 1.	135
Figura 69 - histórico do estoque do material 5, UBS 1.	135
Figura 70 - histórico do estoque do material 4, UBS 2.	136
Figura 71 - histórico do estoque do material 5, UBS 2.	136
Figura 72 - histórico do estoque do material 4, UBS 3.	137
Figura 73 - histórico do estoque do material 5, UBS 3.	137
Figura 74 - histórico do estoque do material 4, UBS 4.	138
Figura 75 - histórico do estoque do material 5, UBS 4.	138
Figura 76 - histórico do estoque do material 4, UBS 5.	139
Figura 77 - histórico do estoque do material 5, UBS 5.	139
Figura 78 - Exemplo de amostragem de dados agrupados cujo um relacionamento funcional ($f(x)$) aproximado é procurado.	146
Figura 79 - Apresentação dos desvios do exemplo dos dados amostrados.	146
Figura 80 -Gráfico da Produção de aço nos EUA.	149
Figura 81 - Classificação por tipo de demanda, extraído de [VIN2006].	150
Figura 82 - Representação da Curva ABC, extraído de [MAR2006].	152
Figura 83 - Representação do gráfico dente de serra, extraído de [VIN2006].	154
Figura 84 - Diagrama de classe Calcular a Lista de Materiais que irão Exceder e Zera.	164
Figura 85 - Diagrama de classe faz os cálculos de Estado Geral do Estoque.	167
Figura 86 - Diagrama de classe - Entrar na sociedades Local e Remota.	167
Figura 87 - Caso de uso que recupera a lista de suprimento de materiais vindas do almoxarifado central.	168
Figura 88 - Diagrama de classe Anotar Lista Entrada de Itens.	169

Figura 89 - Caso de uso de análise do Estado Geral do Estoque.....	170
Figura 90 - Caso de uso de comunicação com a sociedade remota.	171
Figura 91 - Caso de uso das ações de informar o usuário sobre a chegada mensagens remotas.	172
Figura 92 - Caso de uso de calcular Gestão do Estoque.	173
Figura 93 - Diagrama de classe do cálculo de Gestão de Estoque.....	174
Figura 94 - Caso de uso para orientar o usuário sobre o estado geral do estoque.	175
Figura 95 - Diagrama de classe - Verificar se Existe Novas Requisições para Enviar para os Setores.	176
Figura 96 – Detalhes do processo de requisição e doação de medicamentos entre UBSs.....	178

Lista de Tabelas

Tabela 1 – Descrição dos Métodos de Percepção e suas respectivas crenças.....	71
Tabela 2 - Descrição das ações geradas em <i>AgentSpeak(L)</i> e seus respectivos métodos em Java.	77
Tabela 3 - Tabela de Níveis de Risco.	98
Tabela 4 – Tabela do Fator de Segurança (K), extraído de [VIN2006].	156
Tabela 5 - Mensagem remota KQML de requisição de material enviada a sociedade remota através do agente facilitador.	179
Tabela 6 - Mensagem local KQML enviada pelo AGT B ao Sig2m da UBS B sobre a chegada de uma requisição de material.	179
Tabela 7 - Mensagem remota KQML de proposta de doação de medicamentos emitida pelo AGT B ao AGT A.	179
Tabela 8 - Mensagem local KQML enviada pelo AGT C ao Sig2m da UBS C sobre a chegada de uma requisição de material.	180
Tabela 9 - Mensagem remota KQML de proposta de doação de medicamentos emitida pelo AGT C ao AGT A.	180
Tabela 10 - Mensagem local KQML enviada pelo AGT A ao Sig2m da UBS A sobre a chegada de proposta de doação de materiais.	180
Tabela 11 - Envio da mensagem remota de confirmação de aceitação da doação de materiais para a UBS B.	181
Tabela 12 - Envio da mensagem remota de confirmação de aceitação da doação de materiais para a UBS C.	181

Glossário

<i>AGT</i>	<i>Agente Gestor Tutor</i>
<i>AIED</i>	<i>Artificial Intelligence in Education</i>
<i>AM</i>	<i>Agente Mensageiro</i>
<i>BDI</i>	<i>Beliefs, Desires and Intentions</i>
<i>CA</i>	<i>Consumo Anual em Quantidades</i>
<i>CAI</i>	<i>Computed Assisted Instruction</i>
<i>CC</i>	<i>Custo Unitário do Pedido de Compra</i>
<i>CI</i>	<i>Cota Ideal</i>
<i>CM</i>	<i>Consumo Médio</i>
<i>CPA</i>	<i>Custo do Material Armazenado</i>
<i>EGE</i>	<i>Estado Geral do Estoque</i>
<i>EI</i>	<i>Estoque Ideal</i>
<i>Emx</i>	<i>Estoque Máximo</i>
<i>ER</i>	<i>Estoque Real</i>
<i>ES</i>	<i>Estoque de Segurança</i>
<i>EV</i>	<i>Estoque Virtual</i>
<i>FRC</i>	<i>Função de Revisão de Crenças</i>
<i>HTTP</i>	<i>HyperText Transfer Protocol</i>
<i>IAD</i>	<i>Inteligência Artificial Distribuída</i>
<i>IAD-ED</i>	<i>Inteligência Artificial Distribuída aplicada à Educação</i>
<i>ICAI</i>	<i>Intelligent Computer Aided Instruction</i>
<i>IP</i>	<i>Internet Protocol</i>
<i>JVM</i>	<i>Java Virtual Machine</i>
<i>K</i>	<i>Fator de Segurança</i>
<i>KQML</i>	<i>KnowledgeQuery and Manipulation Language</i>
<i>LarCom</i>	<i>Laboratório de Comunicações e Redes de Computadores</i>
<i>LC</i>	<i>Lote de Compra</i>
<i>LCC</i>	<i>Lote de Compra Orientado a Consumo</i>
<i>LVV</i>	<i>Lotes em Véspera de Vencimento</i>

<i>ME</i>	<i>Materiais Excedentes</i>
<i>MME</i>	<i>Média de Materiais Excedentes</i>
<i>MMLVV</i>	<i>Média de Materiais com Lotes em Véspera de Vencimento</i>
<i>MMZ</i>	<i>Média de Materiais Zerados</i>
<i>MVC</i>	<i>Modelo, Visão e Controlador</i>
<i>MZ</i>	<i>Materiais Zerados</i>
<i>NR</i>	<i>Nível de Risco</i>
<i>PP</i>	<i>Ponto de Pedido</i>
<i>PU</i>	<i>Preço Unitário do Material</i>
<i>QC</i>	<i>Quanto Comprar</i>
<i>RMI</i>	<i>Remote Method Invocation</i>
<i>SACI</i>	<i>Simple Agent Communication Infrastructure</i>
<i>SIG2M</i>	<i>Sistema de Gestão de Materiais e Medicamentos</i>
<i>SMA</i>	<i>Sistema Multi-Agentes</i>
<i>SMTP</i>	<i>Simple Mail Transfer Protocol</i>
<i>SOAP</i>	<i>Simple Object Access Protocol</i>
<i>STI</i>	<i>Sistema Tutor Inteligente</i>
<i>STIs</i>	<i>Sistemas Tutores Inteligentes</i>
<i>TCP</i>	<i>Transmission Control Protocol</i>
<i>TR</i>	<i>Tempo de Reposição</i>
<i>UBS</i>	<i>Unidade Básica de Saúde</i>
<i>UBSs</i>	<i>Unidades Básicas de Saúde</i>
<i>UDP</i>	<i>User Datagram Protocol</i>
<i>W</i>	<i>Política de Compra</i>
<i>WSDL</i>	<i>Web Services Description Language</i>
<i>XML</i>	<i>EXtensible Markup Language</i>

Trabalhos Publicados pelo Autor

E. P. Mattos, L. S. Mendes, M. Bottoli. “Uma Proposta de Sociedade de Agentes Inteligentes para a Gestão de Medicamentos nas Unidades Básicas de Saúde”. CBIS - X Congresso Brasileiro de Informática em Saúde "Informática em Saúde e Cidadania". Florianópolis – SC. 14 a 18 de outubro de 2006.

1 Introdução

Esta dissertação concentra-se na área de IADED (*Inteligência Artificial Distribuída Aplicada à Educação*) aplicada à administração de materiais, mais especificamente na arquitetura de STI (*Sistema Tutor Inteligente*). O STI faz parte de uma categoria de sistemas de natureza educacional utilizados como ferramenta de suporte de auxílio ao ensino-aprendizagem.

Um STI é construído com uma estrutura modular que tem por finalidade auxiliar um aprendiz na execução de atividades educacionais. A capacidade de adaptar-se de acordo com as necessidades do aprendiz faz do STI uma arquitetura interessante na construção de softwares educacionais. O STI é modelado usando a arquitetura de agentes cognitivos, o que proporciona maior flexibilidade em sua construção e maior proximidade com o comportamento de processos semelhantes aos do comportamento humano para a realização de tarefas de tutoria para com os aprendizes.

Este trabalho tem por objetivo utilizar a arquitetura de STI como sistema automático de tutoria para sistemas de gestão de materiais. Em particular, a técnica é aplicada para um sistema de gestão da armazenagem e distribuição de medicamentos nas unidades básicas de saúde (UBS) do município de Campinas, SP - Brasil. Para cada UBS, é construído um STI que tem por função realizar o papel de um agente gestor de estoque (Agente Gestor Tutor - AGT), cargo pouco comum na rede municipal de saúde, auxiliando o administrador de cada setor (visto como aprendiz), a realizar tarefas complexas, inerentes à gestão de materiais e medicamentos. Algumas destas tarefas são: estimar um estoque padrão de medicamentos; definir precisamente a quantidade necessária para atender um ciclo de distribuição (cota) de acordo com as suas reais necessidades; prever quando comprar material; prever o quanto comprar de material; alertar quando o estoque atingir o estado de risco (abaixo de um valor crítico); alertar quando ocorrer a necessidade de devolver materiais em estado excedente para o almoxarifado gestor; etc. Isto é possível através de procedimentos matemáticos de gestão de estoque.

Os STI podem ser vistos como organizados em uma sociedade, que deve realizar, em trabalho cooperativo, o processo de requisição, distribuição e redistribuição de materiais entre as UBSs da rede. Com este modelo, busca-se gerar uma certa estabilidade no estoque distribuído.

Este trabalho apresenta como resultado um protótipo de sociedade de STIs, desenvolvido como assistente para a resolução do problema de distribuição de medicamentos. Apresenta-se também um estudo de caso que consiste em uma simulação (correspondente a seis meses de processamento) embasada na arquitetura de uma rede municipal de saúde, com dados autênticos¹ (bases de dados) de cinco UBSs com o mínimo de dois anos de uso.

Os resultados da simulação estão dispostos em tabelas que contêm informações sobre a convergência do estoque real para o ideal, calculada pelo AGT, bem como a repercussão das ações realizadas pelo usuário (aprendiz) auxiliado pelo seu respectivo AGT, diante de diversas situações.

O uso de STI na administração de materiais é uma vertente de pesquisa um tanto diferenciada de sua proposta inicial (construção de ambientes educacionais). Nosso interesse nesta modelagem foi em buscar uma maior proximidade do comportamento do sistema com o usuário para a resolução das tarefas que requerem mão-de-obra especializada. Diante de tais fatos, o emprego dos STI pode ser encarado como um novo paradigma para o desenvolvimento de sistemas de apoio à decisão.

1.1 Motivação

Embora, não se trate de uma rede educacional, o ambiente de uma rede municipal de saúde tem todos os requisitos necessários para o uso de uma sociedade de STIs. Os usuários podem ser vistos como aprendizes, os quais serão auxiliados pelos tutores na complexa tarefa de gerir um estoque, de maneira que não haja escassez ou excesso de material em um determinado ciclo. O material didático utilizado é constituído do próprio estoque de materiais e medicamentos, repositório rico em informação que pode fornecer ao aprendiz dados significativos para o aprendizado, que é tratar o estoque na busca de um estoque ideal às necessidades da rede.

¹ Foram usados dados do sistema da rede de distribuição de medicamentos da Prefeitura Municipal de Campinas, obtidos com autorização.

Estimar um estoque padrão de medicamentos, definir precisamente a quantidade necessária para atender um ciclo de distribuição (cota) para cada UBS, prever períodos sazonais de consumo, monitorar o estoque para averiguar se algum material está se esgotando, entre outras, são tarefas complexas que necessitam de mão-de-obra especializada e de sistemas computacionais que auxiliem na execução destas tarefas.

A constante rotatividade das funções dos usuários nos setores, faz com que o usuário não consiga absorver a cultura de uso do sistema de gestão de forma satisfatória, por este motivo, vale a pena utilizar STIs para que tais usuários sejam constantemente orientados e treinados pelo seu tutor (um STI), dispensando os constantes processos de treinamento de usuários do sistemas gestão.

Por estas razões, acreditamos que o uso de STIs poderá contribuir para aperfeiçoar os sistemas para gestão de materiais, em uma rede de UBSs ligada a uma unidade central de distribuição de materiais e medicamentos, onde o processo de distribuição de materiais ainda é ainda incipiente.

Através da modelagem do Agente Gestor Tutor (AGT) sobre a arquitetura de um STI, é possível executar ações similares às de um profissional humano dotado de estratégias pedagógicas que auxiliem o usuário no monitoramento do estoque, informando as anomalias e dando sugestões sobre como atuar em uma determinada situação crítica.

1.2 Objetivo

O objetivo deste trabalho é modelar uma Sociedade de STIs, para o sistema de gestão de materiais e medicamentos de uma rede de saúde de uma prefeitura municipal. Esta sociedade é projetada para desempenhar tarefas parecidas às de um profissional da área, auxiliando-o na tentativa de tangenciar o estoque o mais próximo possível de suas reais necessidades.

1.3 Disposição dos Capítulos

Os capítulos desta dissertação estão dispostos da forma a seguir.

O Capítulo 1, apresenta a conceituação geral do problema. Este capítulo também apresenta as motivações e objetivos deste trabalho.

O Capítulo 2 apresenta a conceituação geral de Sistemas Tutores Inteligentes e *Agentes*, e é dividido em duas seções: Na primeira apresenta-se o modelo geral dos STIs, além de uma breve descrição sobre seu estado da arte e arquitetura geral. A segunda seção fala sobre *IAD*, o uso da tecnologia de *Sistemas Multi-Agentes* para a construção de sistemas inteligentes e seus elementos, os agentes. É apresentada a sua definição, arquiteturas mais conhecidas e sistemas de protocolo de comunicação entre os agentes.

O Capítulo 3, descreve o *Sistema de Gestão de Materiais e Medicamentos – Sig2m* utilizado pela rede de municipal de saúde de Campinas para a distribuição e controle dos materiais e medicamentos entre seu almoxarifado central e as *Unidades Básicas de Saúde*. Neste capítulo, é apresentada a arquitetura do software, seus módulos de operação, vantagens e limitações. Também são apresentadas algumas propostas de solução para resolver tais limitações.

No Capítulo 4, Abordagem de Sociedades de STI como Instrumento de Auxílio à Gestão de Materiais e Medicamentos, é apresentado o protótipo implementado, a disposição de sua arquitetura separada em camadas, e as tecnologias utilizadas em sua construção.

Finalmente, o Capítulo 5, apresenta os resultados obtidos nas simulações realizadas. Os resultados são analisados, e é discutida a interação do aprendiz (usuário) com o protótipo.

No Capítulo 6, Discussão e Conclusões, contém a conclusão do trabalho produzido. São sumarizadas as contribuições obtidas e descritas as propostas de possíveis trabalhos futuros.

2 Sistema Tutores Inteligentes e Agentes

2.1 Introdução

Este capítulo dividi-se em duas partes. Na primeira, tratada nos itens 2.2 e 2.3, apresentamos a arquitetura de *Sistemas Tutores Inteligentes* (STIs), sistemas destinados ao ensino-aprendizagem. São também ilustrados os principais componentes que compõem sua estrutura, suas vantagens, suas limitações e a sua construção baseada na arquitetura de *Sistemas Multi-Agentes* (SMA).

Na segunda parte, tratada a partir do item 2.4, são apresentados os *Sistemas Multi-Agentes*, subcampo da linha de *Inteligência Artificial Distribuída* (IAD), que têm por principal função resolver problemas de natureza distribuída. São apresentadas também algumas definições sobre agentes, tais como sua arquitetura básica, tipos de agentes, a sua organização social e o protocolo de comunicação mais utilizado.

2.2 Estado da Arte dos STI

A utilização de técnicas de *Inteligência Artificial Distribuída* (IAD) em programas destinados a auxiliar no processo de ensino-aprendizagem tem o seu início na década de 70, com o trabalho de Carbonel [CAR1970], forjando uma nova vertente de pesquisa conhecida como *Artificial Intelligence in Education* (AIED). Carbonel propôs construir sistemas educacionais adaptativos que tinham por objetivo levar em consideração o perfil do aprendiz que estava interagindo, bem como, prover um sistema de tutoria capaz de prestar assistência ao aprendiz sobre as tomadas de decisão e ações perante uma dada tarefa.

O estudo de Carbonel tornou-se um marco sobre o histórico dos STIs. Seu sistema, conhecido como SCHOLAR [CAR1970], foi considerado um padrão, o clássico dos STIs. Seu trabalho objetivava atender ao grupo de programas que possuía conteúdo modelado com um determinado objetivo educacional. Também tinha como objetivo adaptar da melhor forma possível a apresentação deste conteúdo, conforme o perfil de cada aprendiz, ou seja, criar

um sistema que procurava se adaptar ao usuário e que não se comportasse de forma igual com todos os usuários, como era o padrão utilizado na época para os *Computed Assisted Instruction* (CAI). Esta nova modalidade de sistemas passou a ser reconhecida como *Sistemas Inteligentes de Instrução Assistida por Computador*, tradução da sigla *Intelligent CAI* (ICAI).

No grupo dos ICAI surgiram os STIs, sistemas computacionais com modelos de conteúdo instrucionais que especificam o que ensinar e estratégias de ensino que indicam como ensinar [MUR1999]. Trata-se de uma categoria de sistemas de natureza educacional, constituída de uma arquitetura modular para a construção de ferramentas de suporte de auxílio ao ensino-aprendizagem sob a forma de tutoria.

Se considerarmos que estudantes possuem diferentes estilos ou formas de aprendizagem, é importante que um STI seja capaz de fornecer instrução individualizada, ou seja, deve ser capaz de adaptar suas instruções para satisfazer os estilos individuais de cada aprendiz. Desenvolver um STI, modelar um tutor que tenha a capacidade de se adaptar por completo a um aprendiz, é um dos grandes desafios na área de AIED.

No ambiente de sala de aula, o professor obtém retorno do aprendiz de várias maneiras: através do diálogo, ao observar as suas atitudes, pela expressão corporal, etc. Isto possibilita inferir sobre o estado cognitivo do aprendiz, (“termômetro” que mede o seu nível de aproveitamento) e, quando necessário, o tutor (neste caso o professor) pode mudar a sua estratégia de ensino em conformidade com as dificuldades do aprendiz.

Transcrever tais comportamentos para um ambiente computacional é um desafio. Mesmo com o avanço exponencial que o *hardware* e *software* têm sofrido nos últimos anos, estes ainda não atingiram um nível tecnológico capaz de analisar uma expressão corporal de um aprendiz e inferir com absoluta certeza qual abordagem deve ser seguida para oferecer o estímulo certo ao aprendizado.

2.3 Arquitetura de um STI

A Figura 1 apresenta o modelo clássico de arquitetura de um STI, encontrada nos trabalhos de [BOL2002] [GIR1999] [FRA1998] e [OLI1994]. A arquitetura dos STI é composta basicamente pelos seguintes componentes: *Módulo Tutor*, *Módulo de Domínio*, *Módulo do*

Aprendiz e Módulo Interface. Através do módulo interface, o aprendiz interage com o módulo domínio (detém o conteúdo para aprendizagem do aprendiz) acessando seus recursos disponíveis e gerando uma ação no ambiente. As ações do aprendiz são mapeadas como um comportamento que alimentam o módulo do aprendiz. O módulo do aprendiz tem como objetivo fazer uma representação do conhecimento e habilidades cognitivas do aprendiz em um dado momento. Este módulo é caracterizado por possuir dados estáticos e dinâmicos, para que o tutor possa inferir sobre esta representação perante uma dada atividade, e auxiliar o aprendiz em sua resolução, sugerindo caminhos possíveis para resolver uma tarefa, alertando sobre seus pontos críticos e informando sobre o progresso do aprendiz em sua resolução.

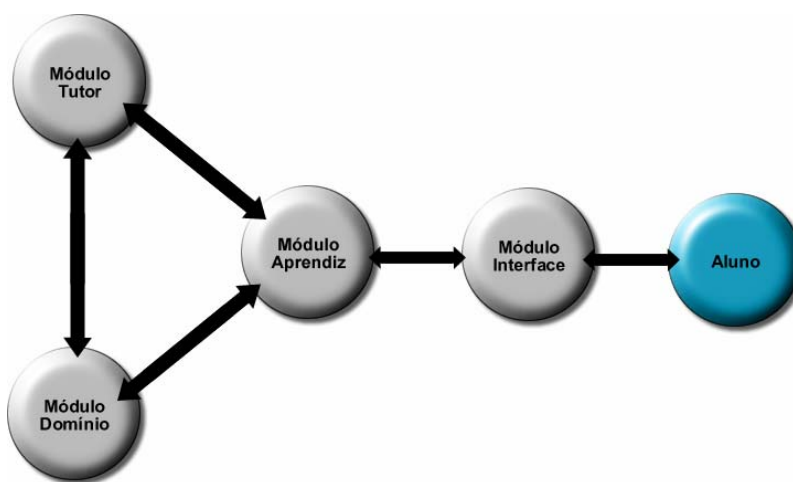


Figura 1 - Arquitetura clássica de um STI.

2.3.1 Módulo do Aprendiz

O módulo do aprendiz tem como objetivo apresentar uma representação do conhecimento e habilidades cognitivas do aprendiz em um dado momento. Este módulo é caracterizado por possuir dados estáticos e dinâmicos para que o tutor possa inferir sobre esta representação, perante uma dada atividade. Desta forma, torna-se possível ao tutor obter um valor qualitativo sobre o conhecimento do aprendiz em relação ao conteúdo abordado. A partir dos módulos do aprendiz e do domínio descrito no item 2.3.3, o tutor poderá decidir qual será a melhor estratégia de ensino a ser utilizada. O módulo do aprendiz

deve ser atualizado de forma dinâmica a partir da analítica cognitiva das ações realizadas pelo aprendiz no ambiente. Desta forma, é necessário que o aprendiz seja constantemente monitorado pelo seu tutor para avaliação de seu desempenho.

Os dados dinâmicos dos aprendizes são construídos a partir de suas ações na resolução de conteúdos ou problemas apresentados pelo tutor. Segundo Giraffa [GIR1999], existem várias técnicas para realizar a construção do módulo do aprendiz. Dentre estas, podemos destacar:

- “Reconhecimento de padrões” aplicado ao histórico das respostas fornecidas pelo aprendiz;
- Comparação da conduta do aprendiz com a de um especialista e verificar os pontos em comum;
- Preferências do aprendiz;
- Objetivos particulares do aprendiz;
- Anotar as ações que o aprendiz costuma esquecer quando interage com o tutor.

Segundo Costa [COS1996], o módulo do aprendiz pode ser representado através de alguns modelos de descrição. Dentre estes, os mais utilizados são:

- Modelo Diferencial – Consiste da comparação da resposta do aprendiz com a base de conhecimento, ou seja, compara-se a performance do “especialista” com a do aprendiz. A modelagem por diferenciação divide o conhecimento em duas classes: aquele que se espera e aquele que não se espera do aprendiz. Nesta modelagem o conhecimento do aprendiz é somente um subconjunto do conhecimento do especialista;
- Modelo de Perturbação – Conhecido como modelo *BUGGY*. Este modelo assume que os erros do aprendiz são decorrentes da concepção errônea ou ausência de algum conceito. Também relaciona o modelo do aprendiz com a base de conhecimento do domínio;
- Modelo de Simulação – Consiste na modelagem do comportamento do aprendiz em determinadas situações. Este modelo torna possível prever o comportamento do aprendiz durante a sessão de trabalho;
- Modelo de Agentes – Tem por finalidade tratar o modelo do aprendiz como um sistema de crenças, desejos e intenções. A interação entre aprendiz e o sistema tutor

é uma interação entre dois agentes inteligentes (ou, pelo menos, dotados de algum comportamento cognitivo). Considerar o aprendiz como um agente implica em considerar o módulo do aprendiz como um modelo de agente. Esta consideração apresentará conseqüências para a estrutura do modelo. O módulo do aprendiz enquanto um modelo de agente, deve incluir, de um modo ou de outro, três componentes: a *base de crenças*, a *base de motivações* e o *modelo de intenções*. Estes componentes podem ser definidos como segue:

- A base de crenças compreende o conhecimento do agente sobre o ambiente (domínio da aplicação). Toda a base de crenças pode ser revisada;
- A base de motivações fornece a representação das informações primárias, e compreende os objetivos a serem atingidos pelo aprendiz;
- O modelo de intenções pode ser pensado como um conjunto de regras de inferência. Cada uma das regras modela um tipo de inferência ou mecanismo de aprendizagem disponível para o agente, podendo atuar tanto sobre a base de crenças como sobre a base de motivações;
- *Modelo Overlay* – Este modelo assume que os erros ou comportamentos anômalos do aprendiz são sempre devidos à ausência de alguma informação presente na base do domínio. No entanto, considera-se que esta não é uma visão completa, pois em algumas situações os comportamentos incorretos acontecem em decorrência da presença de concepções incorretas na mente do aprendiz;

2.3.2 Módulo Tutor

O módulo tutor (ou módulo pedagógico) oferece uma metodologia para o processo de aprendizado. Detém o conhecimento sobre as estratégias e táticas pedagógicas para selecioná-las o mais próximo possível do perfil do aprendiz a fim de auxiliá-lo sobre a resolução de uma dada tarefa. As estratégias e táticas pedagógicas constituem um conjunto de regras e / ou planos para alcançar metas específicas.

Existem várias abordagens de orientação que o tutor pode utilizar na orientação do aprendiz, dentre as quais podemos citar o envio de mensagens (conteúdos diferenciados) e o envio de avisos sonoros (para indicar uma situação crítica, apresentar exemplos de como o

aprendiz deve agir sobre uma situação ou para a apresentação de uma regra). Basicamente, o tutor é composto por um conjunto de estratégias pedagógicas que define a qualidade de um ambiente de aprendizagem.

Conforme Giraffa [GIR1999], as estratégias pedagógicas usadas em um STI possuem as seguintes classificações:

- *Socráticos* - Modelo projetado para induzir o aprendiz a identificar enganos ou interpretações errôneas sobre o conteúdo, através de sequência de perguntas encadeadas baseadas no erro do estudante;
- *Reativos* – O conteúdo reage aos questionamentos do estudante buscando apresentar hipóteses sobre as idéias do estudante e, seqüencialmente, são apresentadas as possíveis implicações de uma ação realizada;
- *Treinamento (Coaching)* – Escolhe-se a forma de ensino mais apropriada a partir um determinado conjunto de regras. O STI tem a função de observar o desempenho do estudante e prover conselhos que o ajudarão a executar uma dada atividade da melhor forma possível. Esta estratégia inclui uma orientação para tarefas novas e interação com o estudante para achar as melhores opções para a resolução de uma dada atividade;
- *Colaborativo (ajuda)* – Este modelo possui menos controle do que o modelo de treinamento. O tutor realiza a conversação com o participante com a finalidade de auxiliá-lo utilizando a sua própria perícia em momentos chave. Neste modelo o tutor poder ser visto como um treinador, mas que interfere em situações críticas em que o aprendiz não consegue solucionar o problema, baseado no número de tentativas feitas na resolução de uma atividade. Este modelo é adequado para ambientes com ênfase em solução de problemas;

2.3.3 Módulo de Domínio

O Módulo de Domínio detém as informações de todo o conteúdo apresentado ao aprendiz. Na íntegra, trata-se do material didático utilizado pelo tutor para instruir o aprendiz.

É um componente chave que define o sucesso de um STI, levando em consideração que será a base de conhecimento do aprendiz. O aprendiz e o tutor realizarão interações sobre o módulo de domínio para a construção do conhecimento. O módulo de domínio pode ser dinâmico, conforme o processo de desenvolvimento da aprendizagem.

2.3.4 Módulo Interface

Este módulo corresponde à camada responsável por gerar a comunicação entre o tutor e o aprendiz. Sua complexidade pode abranger grandes variações. Abrange desde as janelas mais simples de diálogo até mesmo sistemas de reconhecimento de voz, linguagem natural e sistemas de realidade virtual que proporcionam ao estudante a imersão total no ambiente.

2.4 STI e Sistemas Multi-Agentes

A modelagem de STI combinada com *Inteligência Artificial Distribuída* (IAD), mais especificamente com *Sistemas Multi-Agentes* (SMA), tem sido utilizada há alguns anos, conforme pode ser observado nos trabalhos de Mousalle *et al* [MOU1996], Costa [COS1996], Giraffa [GIR1999] e Bolzan e Giraffa [BOL2002]. Bolzan apresenta um comparativo entre alguns trabalhos realizados com esta arquitetura. A combinação de SMA e STI trouxeram uma série de contribuições para a construção de sistemas de ensino e aprendizagem, tais como:

- **Distribuição do conhecimento:** O conhecimento do tutor pode ser distribuído para vários tutores, que tratam este conhecimento com o seu próprio conjunto de crenças, desejos e intenções. Assim, gera-se tutores especializados em cada assunto abordado, o que possibilita maior qualidade sobre as estratégias pedagógicas;
- **Divisão de responsabilidades:** A arquitetura STI pode ser dividida em camadas, representadas por agentes (ou uma sociedade de agentes). Ela proporciona a divisão das responsabilidades de uma entidade para várias, como por exemplo, uma sociedade de agentes destinada a reger a camada de comunicação do sistema, outra para seleção de estratégias pedagógicas, etc;

- **Autonomia das entidades:** Agentes de um modo geral são autônomos. Na maior parte do tempo, não necessitam de intervenção humana. Não possuem uma autoridade centralizadora destinada a realizar o controle de suas ações e podem reagir às ações do aprendiz de forma coerente;
- **Modularidade do sistema:** Agentes são entidades atômicas e independentes, o que torna mais fácil serem removidos e adicionados a um sistema computacional. Este é um ponto positivo quando estamos falando de sistemas educacionais, que possui elementos dinâmicos como adição e remoção de novos módulos educacionais, personagens, etc;
- **Cooperação no aprendizado do aprendiz:** Uma das vantagens dos SMA é a possibilidade de construção de sistemas cooperativos em que cada agente se relaciona com os demais, a fim de solucionar um determinado problema complexo. Esta característica peculiar pode ser adotada nos STIs construídos sob agentes, que cooperam entre si para melhor auxiliar o aprendiz em seu aprendizado;

2.4.1 Inteligência Artificial Distribuída (IAD)

De acordo com Weiss [WEI1999] “A resolução de problemas distribuídos é o nome aplicado ao subcampo da inteligência artificial distribuída (IAD) que enfatiza o uso de agentes para trabalharem juntos em problemas que requerem um esforço coletivo”. Basicamente, um sistema de IAD é formado pelos seguintes elementos conforme Alvarez *et al* [ALV1997]: *Agentes* - entidades ativas do sistema; *Sociedade* - conjunto formado por agentes; *Ambiente* - conjunto formado pelos objetos do sistema (objetos passivos); *Interação* - troca de informações entre os agentes (podem ser trocas diretas, através de comunicação explícita, ou pelo ambiente); e *Organização* – que garante que os agentes farão aquilo para que foram projetados.

2.4.2 Agentes Inteligentes e sua Definição

São várias as definições sobre agentes [MUR1999] [WEI1999] [BOR2001] [WOO2005], no entanto, não se pode afirmar que qualquer uma delas forme uma visão completa do conceito, de forma que não se definiu ainda um padrão.

Russell [RUS1995] define agente como uma entidade que é capaz de perceber eventos em seu próprio ambiente, através do uso de sensores, e agir sobre ele através de seus atuadores. Genesereth [GEN1994] define que praticamente qualquer processo que tem a habilidade de se comunicar pode ser denominado agente. Entretanto, existem algumas definições propostas por Wooldridge [WOO2005] que são as mais difundidas no mundo dos agentes. Wooldridge relata que existem dois níveis de noções sobre agentes, conforme descrito a seguir.

Noção fraca: o termo agente é denotado tanto para o *hardware* quanto para o *software* que possuam as seguintes propriedades:

- Autonomia: os agentes operam sem a intervenção direta do ser humano e / ou outra entidade. Possuem todo tipo de controle sobre os seus estados internos.
- Sociabilidade: Agentes interagem com outros agentes (podendo ser humanos ou máquinas) mediante algum tipo de linguagem de comunicação compreensível por ambas as partes.
- Reatividade: agentes percebem o ambiente e reagem instintivamente em tempo às mudanças ocorridas por ele.
- Pró-atividade: agentes não respondem apenas às mudanças no seu ambiente, eles ainda podem exibir um comportamento orientado a objetivo. Possui autonomia de tomar decisões.

Noção forte: uma visão mais especializada atribuída, principalmente, pelos estudiosos envolvidos em IA. Esta noção define sistemas computacionais que, além de possuírem as características citadas acima, são construídos baseados nos conceitos do comportamento humano, como estados mentais compostos pelas crenças, desejos e intenções.

Pode-se enfatizar de forma genérica que um agente é uma entidade (real ou não) que está inserida em um contexto (em um ambiente), e que possui comportamento autônomo, isto é, capacidade de perceber, agir, deliberar e comunicar-se com outros agentes de forma a alterar o estado do ambiente.

Vários autores citaram em seus trabalhos algumas características comuns e de certa relevância sobre o perfil de um agente. Segue abaixo algumas delas relativas à capacidade do agente (“capacidade de”).

- **Perceber:** o agente é capaz de perceber alterações no ambiente.
- **Agir:** as alterações no ambiente são feitas em virtude das ações que os agentes realizam na tentativa de atingir seus objetivos, ou seja, transformam o seu estado de mundo em um estado apropriado.
- **Comunicar:** o agente pode se comunicar com outros agentes da sociedade que compartilham o mesmo ambiente. A comunicação pode ser caracterizada como uma ação realizada pelo agente. É um poderoso artifício para realizar a troca de informações entre os agentes da sociedade e até mesmo coordenar e / ou cooperar entre si para atingir um objetivo pretendido.
- **Mover:** o agente possui a capacidade de se mover intencionalmente de um local a outro.
- **Representação do ambiente:** o agente possui uma representação simbólica e explícita daquilo que acredita ser verdadeiro sobre o ambiente e também sobre os demais agentes que compartilham deste mesmo ambiente.
- **Motivar sobre um objetivo:** genericamente os agentes são autônomos por esta característica peculiar, é importante que se tenha uma representação dos desejos e objetivos, ou seja, ter uma representação de estados do ambiente que o agente almeja alcançar. Como consequência, o agente age sobre o ambiente por iniciativa própria para satisfazer tais objetivos.
- **Deliberar:** mediante a motivação e a representação sobre o estado atual do ambiente, o agente tem por função selecionar dentre os estados do ambiente, quais têm grande possibilidade de ocorrer no futuro: Estes possivelmente tornar-se-ão objetivos a serem atingidos.
- **Raciocinar e aprender:** os agentes também podem ser dotados de raciocínio lógico e aprendizado através das técnicas de inteligência artificial clássica, como aprendizagem por reforço, emprego de redes neurais artificiais, etc. O raciocínio e aprendizagem podem trazer desempenhos significativos sob o aspecto da deliberação.

2.4.3 Arquiteturas de Agentes

Os agentes são classificados em várias arquiteturas, dentre elas podemos destacar as arquiteturas: reativa e cognitiva. Os agentes reativos são mais simples, reagem instintivamente perante as modificações realizadas no ambiente, formado pelo par Estímulo-Resposta (Ação-Reação). São baseados em modelos de organização biológica (como peixes, formigas e cupins).

A Figura 2 apresenta o funcionamento de um agente reativo, modelo proposto por Wooldridge [WOO1999]. Nesta arquitetura, a ação é uma função direta dos sensores do agente. O agente faz a percepção do ambiente, atualizando o seu estado interno e em seguida é selecionada uma ação, em função do seu estado. Este ciclo é executado até que seu objetivo tenha sido alcançado.

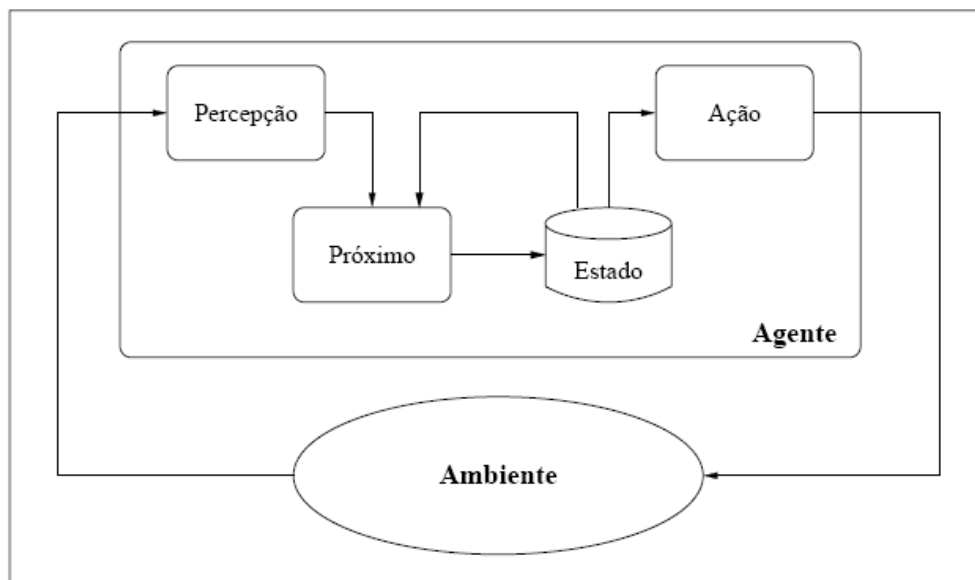


Figura 2 - Modelo geral de agente proposto por [WOO1999].

Ao contrário da arquitetura reativa, a arquitetura cognitiva é mais complexa: Pertencente ao ramo da Inteligência Artificial Simbólica detém a capacidade de realizar tarefas inteligentes sob um sistema de planejamento baseado em conceitos da psicologia cognitiva²

² Psicologia cognitiva é a ciência, um ramo da psicologia que estuda a cognição, o processo mental que hipoteticamente está por detrás do comportamento do ser humano.

conhecida como estados mentais (como crenças, desejos, intenções). Nesta arquitetura os agentes constroem um conjunto de planos e ações para atingir um objetivo pretendido [WOO1999], [WOO2000] e [WEI1999].

A Figura 3 apresenta a arquitetura de agentes cognitivos fundamentada em estados mentais, também conhecida como arquitetura *Beliefs, Desires e Intentions* (BDI) proposto por Bratman [BRA1988].

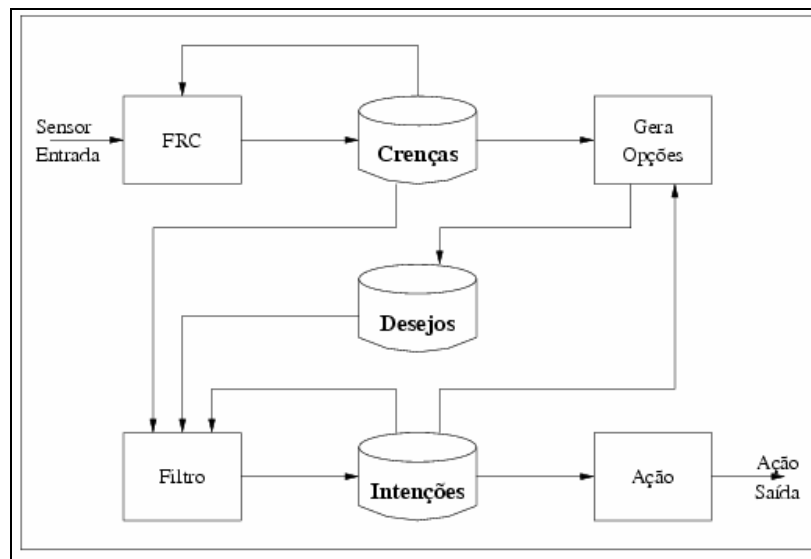


Figura 3 - Arquitetura BDI Genérica (adaptado de por [WOO1999]).

Nesta arquitetura, as *crenças* representam a visão acerca do ambiente (seu estado) e também dos agentes envolvidos na sociedade. Os *Desejos* representam o estado que o agente aspira atingir, ou seja, uma representação do estado que o agente quer que passe a ser verdadeiro no ambiente. Um *desejo* é composto por um conjunto de objetivos vinculados entre si. As *Intenções* são seqüências de ações específicas que um agente se compromete a executar para atingir determinados objetivos. A Função de Revisão de Crenças (*FRC*) tem a função de verificar as crenças anteriores a serem atualizadas, resultando em um novo estado do ambiente a partir das percepções do agente. Atingindo um novo estado do ambiente, é possível que novas alternativas de estados a serem atingidos fiquem disponíveis. O componente *Gera Opções*, apresentado na Figura 3, tem por função verificar quais são estas novas alternativas de estado a serem atingidas. Este processo é realizado através das

crenças do agente e das intenções do agente. A atualização dos objetivos pode ser realizada de duas formas: através da observação do ambiente que pode determinar novos objetivos a serem alcançados e pela execução das intenções.

O *Filtro* é responsável por realizar a atualização das intenções como resultado das crenças, os desejos e das intenções já existentes. Este processo é conhecido como deliberação [HUB2004].

2.5 A Linguagem AgentSpeak(L)

A linguagem *AgentSpeak(L)* é uma linguagem projetada para a construção de agentes BDI proposto por Rao [RAO1992]. Sua construção é baseada em sistemas de planejamento reativo, constituem a classe de sistemas que estão permanentemente em execução reagindo a eventos que acontecem no ambiente em que estão situados, através da execução de planos de uma biblioteca de planos [GEO1987], [BOR2003] e [BOR2005].

A linguagem é uma extensão da programação em lógica para a arquitetura de agentes BDI. Atualmente a linguagem *AgentSpeak(L)* representa um modelo abstrato para a programação de agentes e tem sido a abordagem predominante na implementação de agentes inteligentes ou “racionais” [WOO2000]. Agentes programados em *AgentSpeak(L)* contém um conjunto de crenças titulado como *base de crenças* e um *conjunto de planos*. Uma crença é um predicado de primeira ordem, que é composto por um ou mais termos, como por exemplo, a crença *fazer_reserva (cadeira_desocupada, primeira_fila,...)* possui o predicado *fazer_reserva* e seus termos são: *cadeira_desocupada, primeira_fila*.

Em *AgentSpeak(L)*, existem dois tipos de desejos (objetivos): *objetivos de realização* e *objetivos de teste*. Os objetivos de realização e teste são compostos por predicados, assim como as crenças, mas contém operadores prefixados “!” e “?” respectivamente. Objetivos de realização expressam que o agente quer alcançar um estado no ambiente onde o seu predicado associado é verdadeiro, que iniciam a execução de outros planos. Um objetivo de teste retorna a unificação do predicado de teste com uma crença do agente, ou uma falha caso não seja possível a sua unificação.

Um *evento ativador* define quais eventos podem iniciar a execução de um plano. Um evento pode ser interno, quando gerado pela execução de um plano em que um *objetivo*

específico precisa ser alcançado, ou externo, quando gerado pelas atualizações de crenças feitas pela percepção do ambiente. Os *eventos ativadores* são relacionados com a adição e remoção de crenças e objetivos no agente, conhecida como adição e remoção de atitudes mentais. A adição e remoção das atitudes mentais são os operadores prefixados '+' e '-' adicionados nas crenças e nos objetivos do agente.

Os planos contêm *ações básicas* de um agente é capaz de executar em seu ambiente. As ações são definidas por predicados diferenciados conhecidos como símbolos de ação usados para distinguir as ações de outros predicados contidos no código. Um plano é formado por um evento ativador que representa o seu propósito, seguido por um conjunto de crenças representando um *contexto*, conforme apresentado pela Figura 4. Um plano passa a ser *aplicável* quando o contexto é uma consequência lógica do conjunto de crenças do agente no momento em que o evento é selecionado pelo agente.

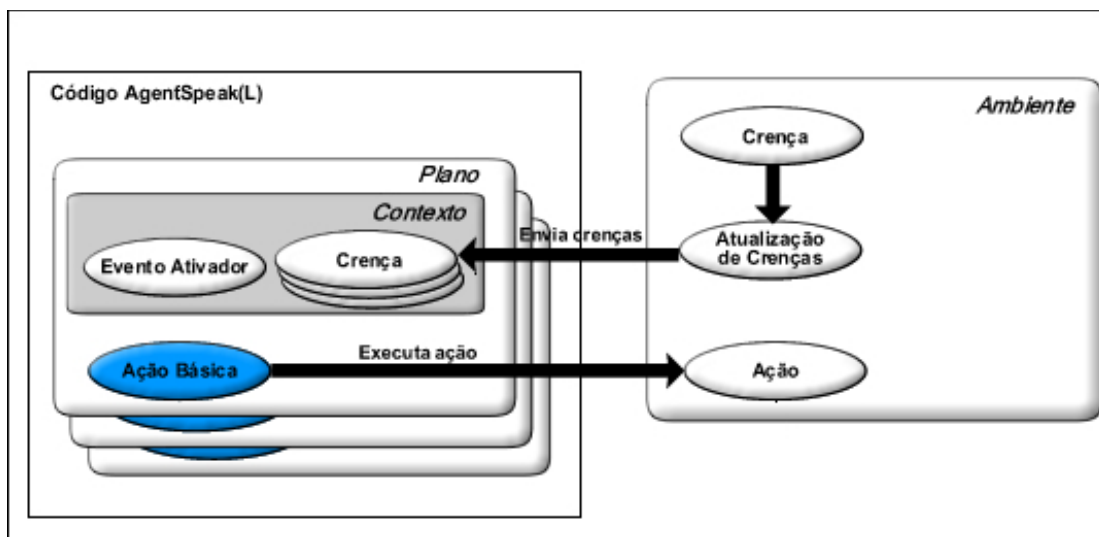


Figura 4 - Código AgentSpeak(L) – estrutura de um plano.

A Figura 5 apresenta um exemplo em *AgentSpeak(L)*, que contém um conjunto de planos para um agente assistir uma peça de teatro. O primeiro plano constitui o anúncio de uma peça de teatro (definido pelo termo: *Peca*) no endereço (*Endereco*), correspondendo a adição da crença *teatro(Peca, Endereco)*, gerada na percepção do ambiente. Se o agente gostar da peça de teatro, então, ele terá como objetivo fazer a reserva de ingressos para a peça, disparado um evento interno definido pelo objetivo *comprar_ingressos(Peca,*

Endereco). O segundo plano tem por função realizar o processo de compra de ingressos. Se a linha telefônica não estiver ocupada (representada pela crença: \neg ocupado(telefone)), o agente realiza a ação básica de ligar para o teatro (ação ligar(Endereco)), seguindo o protocolo de compra dos ingressos (indicado por ...), e que termina com a execução de um subplano para a escolha de assentos no teatro.

```
+teatro(Peca, Endereco) : gostar(Peca)
<- !comprar_ingressos(Peca, Endereco).

!comprar_ingressos(Peca, Endereco):  $\neg$ ocupado(telefone)
<- ligar(Endereco);
...
!escolher_acentos(Peca, Endereco).
```

Figura 5 - Exemplo de código AgentSpeak(L), adaptado de [HUB2004].

2.6 Sistemas Multi-Agentes (SMA)

Sistemas Multi-Agentes (SMA) é um conceito de IAD adotado para o desenvolvimento de sistemas computacionais a partir dos agentes. Os agentes são organizados em *sociedade* e interagem para resolver problemas de natureza distribuída emergindo um comportamento complexo e inteligente [HAL1997] através de três modalidades de interação: cooperativa, colaborativa ou competitiva entre os agentes.

O trabalho colaborativo, os agentes agem em prol de um objetivo comum, eles trabalham em conjunto para cumpri-lo. O trabalho cooperativo, os agentes interagem entre si, mas cada agente possui os seus próprios objetivos, ou seja, não existe um objetivo único para todo o grupo. Já o trabalho competitivo, os agentes interagem entre si, entretanto possuem objetivos conflitantes ocorrendo uma competição entre os agentes, como por exemplo, uma busca por um recurso em que um agente pode ser contemplado e outro não.

Com a abordagem de uso de SMA como metodologia de desenvolvimento de sistemas, possibilitou construir aplicações dotadas das seguintes características [ALV1997]:

- Os agentes são concebidos independentemente de um problema particular;

- A interação entre os agentes não é projetada em seu desenvolvimento, busca-se definir protocolos que possam ser utilizados em situações genéricas;
- As tarefas para solucionar um dado problema podem ser decompostas pelos próprios agentes que realizam a sua administração;
- Não existe um controle centralizado da resolução do problema.

Desta forma, são identificadas algumas vantagens com relação ao uso de SMA, tais como:

- Viabilizam sistemas adaptativos e evolutivos: o SMA tem capacidade de adaptação a novas situações, como a possibilidade de eliminar ou incluir novos agentes ao sistema e mudanças em sua organização.
- Arquitetura emergente para a modelagem de sistemas complexos e distribuídos em que o conhecimento está distribuído, o controle é distribuído, os recursos estão distribuídos. Na modelagem de sistemas, existe a decomposição do problema e a atribuição dos subproblemas aos agentes permitindo um alto nível de abstração e independência entre as partes do sistema [JEN1998].
- Os SMA são operáveis em ambientes heterogêneos e distribuídos: agentes com arquiteturas diferentes funcionam em plataformas diferentes, distribuídas em uma rede de computadores que podem interagir (cooperar, colaborar ou competir) na resolução de problemas. Isto permite o uso das potencialidades particulares de cada arquitetura otimizando o desempenho do sistema.
- Permite conceber sistemas abertos: os agentes podem migrar entre sociedades, ou seja, agentes podem sair e entrar em sociedades, mesmo que desenvolvidos por projetistas e objetivos distintos, possibilitando a construção de sistemas abertos que geram comportamentos inteligentes.
- SMA permite a implementação de ambientes não estáticos: Os SMA permitem a implementação de ambientes em que é composto por partes não-estáticas que podem ser modelados com outros agentes.

2.6.1 A evolução dos SMA

A evolução dos SMA é constituída de três estágios. O primeiro estágio um agente único passou a ter a capacidade de perceber outros agentes que fazem parte do ambiente, agentes que constituem parte do mundo, possibilitando o agente observar e prever o comportamento dos outros agentes para tomar uma decisão. Neste momento, as decisões do agente deixam de ser baseadas apenas em um dado obtido através das percepções do ambiente, passando também a ser fundamentada nas ações da sociedade.

O segundo estágio de evolução de SMA, os agentes passam a se comunicar entre si. Os agentes reconhecem aos outros não mais como parte do ambiente, e sim como uma entidade que tem a capacidade de se comunicar. No terceiro estágio de evolução, os agentes passam a colaborar e cooperar na realização de uma tarefa.

2.6.2 Tipos de SMA

Existem basicamente duas arquiteturas de SMA [HUB2004] [GIR1997] [WEI1999] [WOO1999] conforme descrito a seguir.

- Sistemas Multi-Agentes Cognitivos (SMAC): sociedade composta de agentes cognitivos. Geralmente a sociedade é composta por poucos agentes devido a sua complexidade.
- Sistemas Multi-Agentes Reativos (SMAR): sociedade composta por agentes reativos. A sociedade é composta por muitos agentes.

Atualmente, para a resolução de tarefas, tem-se adotado a fusão dos SMAC e SMAR denominado SMA híbrido, o qual é uma classe de SMA que possui reatividade e proatividade. Conforme Wooldridge [WOO1999], os agentes estão dispostos sob uma hierarquia de camadas, um modelo organizacional de sociedade, composta por agentes cognitivos, que por sua vez, compõem a classe coordenadora, e reativos, que representa a classe operária. Os agentes coordenadores são responsáveis por delegar tarefas para os operários resolverem, este mecanismo é conhecido como *mecanismos de coordenação centralizada*. Existe um segundo mecanismo de coordenação chamado de *mecanismo de coordenação distribuída*, onde todos os agentes estão no mesmo nível hierárquico de trabalho, não

havendo as classes coordenadora e operária. Neste mecanismo, todos os agentes coordenam, executam ações e interagem entre si para realizarem uma tarefa por meio do trabalho cooperativo, colaborativo ou competitivo.

2.7 A Linguagem de Comunicação entre Agentes KQML e o Protocolo de Interação Mecanismo de Contrato Orientado a Agentes Facilitadores

Na resolução de um problema complexo, de natureza distribuída, em que se impõe a utilização de agentes, é de extrema importância que agentes da sociedade se comuniquem a fim de realizarem a troca de informações para sincronizarem o seu estado geral. Em uma sociedade de agentes, a comunicação possui um papel central, e é dela que emergem as ações dos agentes, seja de cooperação, coordenação ou competição (através da negociação entre eles).

2.7.1 Linguagem de comunicação entre agentes (ACL)

A linguagem de comunicação entre agentes (tradução de *Agent Communication Language* - ACL) é baseada na linguagem humana, em que se utiliza a Teoria dos Atos Comunicativos (conhecido também como atos da fala) para se comunicarem. A teoria dos atos comunicativos é uma derivação dos atos lingüísticos da comunicação humana com uma linguagem, que trata a comunicação entre falantes de uma linguagem (fluxo das mensagens) como ações. Ações que são realizadas por agentes e que não diferenciam das demais para indicar as suas intenções.

Na linguagem humana, os atos comunicativos são interpretados a partir da mensagem e do seu contexto, possibilitando identificar se a mensagem trata-se de um comando, um pedido, uma pergunta ou apenas uma informação. Entretanto, na linguagem de comunicação de agentes é necessário deixar de forma explícita o ato comunicativo relacionado à mensagem para que o agente consiga interpretá-lo gerando uma divisão da mensagem em dois níveis hierárquicos: ato comunicativo e seu conteúdo, conforme a Figura 6. O conteúdo da mensagem é expresso através de uma linguagem de conteúdo (*Context Language* - CL),

nele deve ser codificado o conhecimento que se deseja compartilhar com o destinatário da mensagem.

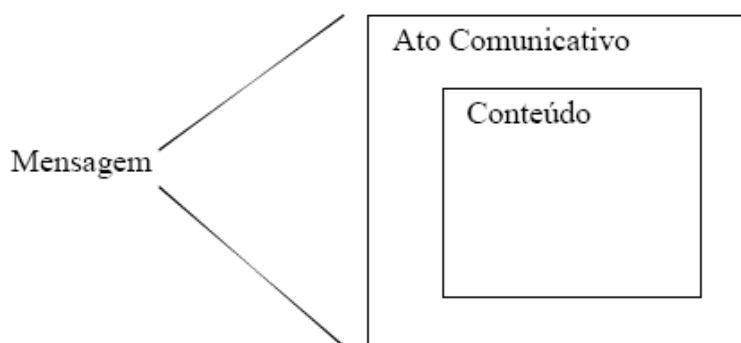


Figura 6 – Estrutura de uma mensagem de um agente.

2.7.2 A Linguagem de Comunicação entre Agentes KQML

Em meados da década de 90 foi fundada o *Knowledge Sharing Effort* (KSE) um consórcio entre empresas, universidades e centros de pesquisa com o propósito de prover uma padronização para a troca do conhecimento representado entre sistemas de informação autônomos [FIN1992], definindo uma linguagem de representação de conhecimento universal que pudesse servir como uma linguagem intermediária entre textos de diferentes linguagens e também prover a interação em tempo real entre sistemas para a troca de conteúdo. Assim, o KSE gerou dois protocolos, descritos a seguir:

Knowledge Interchange Format (KIF): é uma linguagem desenvolvida para servir como uma CL, para realizar a representação do conhecimento sobre um modelo de domínio em particular. Considerado como um padrão para o intercâmbio de conhecimento entre diferentes sistemas. É utilizado como representação da parte do conteúdo de uma mensagem KQML.

Knowledge Query and Manipulation Language (KQML): é uma linguagem desenvolvida como uma ACL para a comunicação entre agentes que possui um formato de mensagem, e contém um protocolo de gerenciamento de mensagens, baseado na teoria dos atos da fala.

Sua linguagem é composta por *performativas*³ que regem as interações entre os agentes, considerado um dos protocolos de comunicação entre agentes mais difundidos no mundo dos SMA [FIN1992], [FIN1994a] e [FIN1994b].

A estrutura básica de uma mensagem KQML, representada pela Figura 7, é composta respectivamente pela performativa, o nome do agente remetente, o nome do agente destinatário, a linguagem empregada de comunicação, a sua ontologia e finalmente o conteúdo da informação na íntegra.

```
performativa  (  :sender      <word>
                  :receiver   <word>
                  :language    <word>
                  :ontology    <word>
                  :content     <expression>  )
```

Figura 7 - Estrutura básica de uma mensagem KQML, extraído de [HUB2003].

Hübner [HUB2004] e Wooldridge [WOO1999] relatam as principais performativas da linguagem KQML agrupadas por categorias. Na descrição abaixo são representados dois agentes: S um agente que envia a mensagem (*sender*) e R um agente que recebe a mensagem (*receiver*).

Performativas de Informação

- **tell**: S informa a R que a natureza do conteúdo da mensagem enviada é verdadeira, ou seja, o conteúdo está na base de conhecimento de S.
- **deny**: S informa a R que a natureza do conteúdo da mensagem enviada não é verdadeira.
- **untell**: o conteúdo enviado não está na base de conhecimento de S.

Performativas de Consulta

- **ask-if**: S quer saber se o conteúdo desta mensagem é verdadeiro para R.

³ Performativas: referem-se à locução que indica uma determinada intenção. Neste contexto, trata-se da intenção de se comunicar do agente.

- **ask-all:** S deseja todas as instâncias da mensagem que são verdadeiras para S.

Respostas Básicas

- **error:** S indica a R que não recebeu a mensagem enviada anteriormente.
- **sorry:** S diz a R que compreende a sua mensagem, no entanto, não pode prover uma resposta satisfatória.

Performativas de Bases de Dados

- **insert:** S pede a R para acrescentar o conteúdo da mensagem na base de conhecimento de R.
- **delete:** S pede a R para remover o conteúdo da mensagem da base de conhecimento de R.

Definição de Capacidades

- **advertise:** S anuncia a R (facilitador) para que saiba que S pode e processará mensagens que estão no conteúdo desta.

Performativas de Efetuação

- **achieve:** solicita que o conteúdo a ser processado da mensagem deve ser um valor verdadeiro.
- **unachieve:** requisição de reversão de um *achieve* enviado previamente.

Performativas de Rede

- **register:** S anuncia para R (facilitador) sua existência mediado pelo nome simbólico associado com seu endereço físico.
- **unregister:** requisita o cancelamento do *register* feito anteriormente.
- **transport-address:** anuncia um novo endereço físico na rede.
- **forward:** comando de encaminhamento de uma mensagem para um outro agente através do campo “:to”.
- **broadcast:** S solicita a R enviar a mensagem a todos os agentes conhecidos de uma sociedade.

Performativas de Facilitação

- **broker-one:** **S** pede a **R** para executar o conteúdo desta mensagem, quando o conteúdo se tratar de uma expressão.
- **recommend-one:** **S** requisita a **R** (facilitador) para indicar um agente que possa processar o conteúdo da mensagem.

Ilustração de uso do protocolo KQML

Uma melhor compreensão sobre o uso do protocolo KQML pode ser obtida através do exemplo abaixo. O presente exemplo, ilustrado pela Figura 8, enfatiza as funções de anúncio de uma habilidade, envio e recebimento de mensagens KQML entre agentes. O exemplo emprega uma sociedade com três agentes que desempenham os seguintes papéis:

- Agente A (sender): anuncia uma habilidade ao facilitador.
- Agente B (facilitador): realiza o papel de facilitador da sociedade.
- Agente C (receiver): requisita uma determinada tarefa à sociedade.

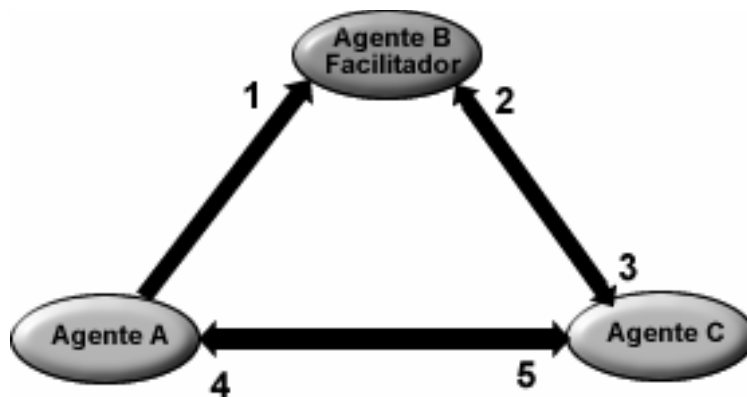


Figura 8 - Exemplo de uso do protocolo KQML.

Anúncio de uma habilidade ao agente B (Facilitador)

O agente A (*sender*) envia uma mensagem ao agente B (*receiver*), suposto facilitador, anunciando que pode processar mensagens do tipo especificado (somar), representado pelo

passo 1 da Figura 8. O conteúdo apresentado da mensagem refere-se a uma outra mensagem em KQML. Esta mensagem possui uma identificação para ser utilizada como tratamento da resposta, através do termo: *reply-with id1*, conforme apresentado abaixo.

```
(advertise
:sender A
:receiver B
:reply-with id1
:language KQML
:ontology kqml-ontology
:content (ask-if
:sender B
:receiver A
:language algebra
:ontology matematica
:content "somar(X,Y)"))
```

Consulta ao facilitador sobre um agente que possa atender à requisição

No trecho KQML em questão, o agente C requisita ao facilitador qual agente pode atender a sua requisição, passo 2 da Figura 8. Abaixo é apresentada a mensagem KQML enviada ao agente facilitador.

```
(recommend-one
:sender C
:receiver B
:in-reply id2
:in-reply-to id1
:language algebra
:ontology matematica
:content "somar(X,Y)")
```

O facilitador (agente B) responde ao agente requisitante (agente C) enviando uma mensagem cujo conteúdo é a identificação do agente que pode atender a sua requisição, neste caso o agente A, passo 3 da Figura 8.

```
(tell
:sender B
:receiver C
```

```
:in-reply id3  
:in-reply-to id2  
:language algebra  
:ontology matematica  
:content "A")
```

Finalmente ocorre a interação entre os agentes C e A, passo 4 da Figura 8. O agente C envia uma mensagem que tem como conteúdo uma expressão matemática, ilustrado pela mensagem KQML abaixo.

```
(tell  
  :sender C  
  :receiver A  
  :in-reply id1000  
  :in-reply-to id1001  
  :language algebra  
  :ontology matematica  
  :content "somar(100,600)")
```

Efetivamente o agente A responde ao agente C enviando uma mensagem tendo como conteúdo o resultado do processamento, fim do ciclo de comunicação entre os agentes, passo 5. Apresentado abaixo.

```
(tell  
  :sender A  
  :receiver C  
  :in-reply id1001  
  :language algebra  
  :ontology matematica  
  :content "700")
```

2.8 O Protocolo de Interação entre Agentes – Mecanismo de Contrato Orientado a Agentes Facilitadores

Além de possuir um protocolo de comunicação, em um SMA é necessário que exista um protocolo que faz a orquestração da comunicação, ou seja, um protocolo de interação entre agentes de forma a sincronizar o fluxo de informação.

Um dos protocolos de interação entre agentes mais conhecidos é o protocolo de *rede de contratos* [WEI1999]. O protocolo de redes de contratos é utilizado para resolver problemas com o trabalho cooperativo entre os agentes. Este conceito parte da idéia de encontrar um agente mais apropriado para realizar uma dada tarefa. A Figura 9 apresenta os passos básicos deste protocolo. Um agente administrador é responsável por anunciar uma tarefa aos agentes contratantes enviando uma mensagem a todos os agentes da sociedade (passo 1 da figura). Em seguida, os agentes contratantes interessados anunciam o seu interesse enviando ao agente administrador a sua resposta, representado pelo passo 2. O agente administrador, por sua vez, seleciona o contratante mais apropriado fazendo uma comunicação privada para a resolução da tarefa, passo 3. Sob a perspectiva do agente administrador os processos identificados são:

- Anúncio das tarefas que necessitam serem realizadas;
- Receber e avaliar os candidatos a contratação;
- Selecionar um contratador mais apto a efetuar a tarefa;
- Receber e sintetizar os resultados;

Sob a perspectiva dos contratadores os processos identificados são:

- Receber anúncios de tarefas;
- Avaliar a capacidade de resolver uma tarefa;
- Responder às mensagens do administrador;
- Realizar a tarefa caso seja contratado a fazê-la;
- Reportar os resultados ao administrador;

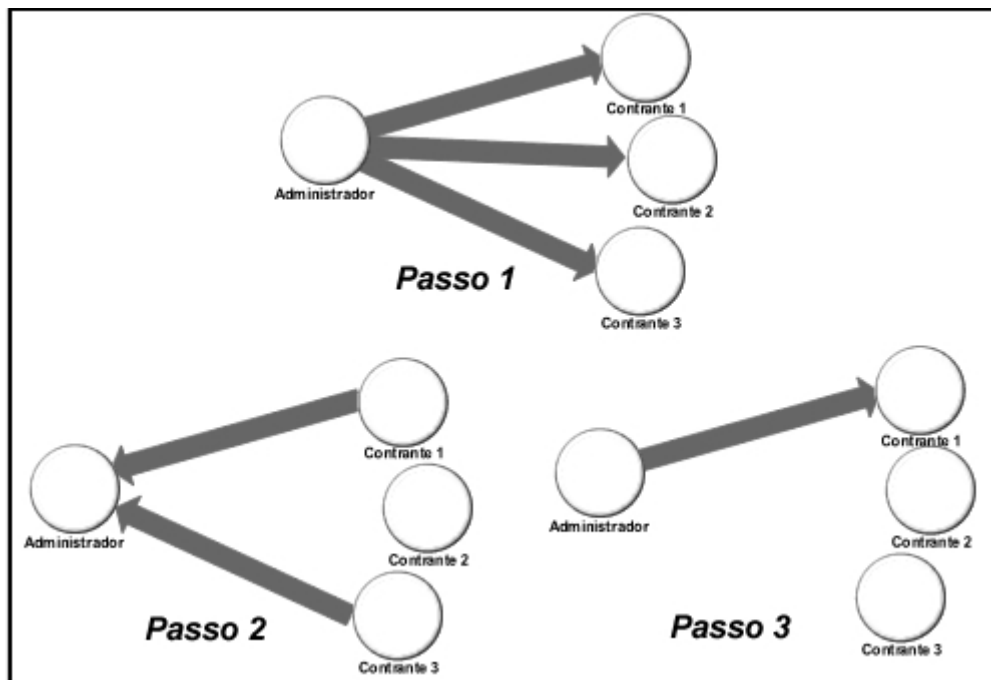


Figura 9 - Protocolo de interação de rede de contrato.

O protocolo de rede de contrato é modelado sob uma arquitetura de comunicação baseada em agentes *facilitadores*, cujo, controle de comunicação entre os agentes é coordenado por um *agente facilitador*, agente complexo que tem por finalidade armazenar informações de localização lógica e habilidades dos agentes de uma sociedade, conforme Hübner [HUB2003]. Toda vez que entra na sociedade, um determinado agente deve anunciar ao agente facilitador a sua entrada e publicar as suas funcionalidades (passos 1 e 2 da Figura 10). Por exemplo, quando o agente A pretende se comunicar com o agente B e não possui o seu endereço, este recorre ao agente facilitador (passo 3) para solicitar o endereço na rede do agente B (passo 4), e então realizar a comunicação (passo 5).

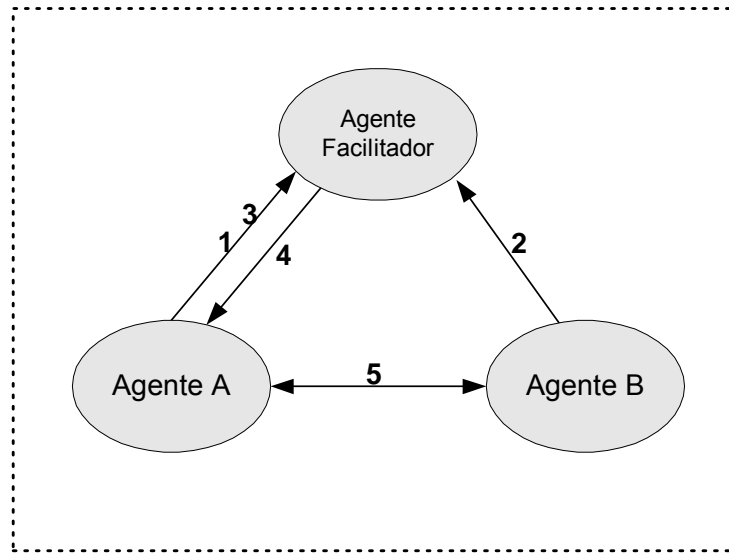


Figura 10 - Sociedade de agentes intermediada pelo agente facilitador.

2.9 Bibliografia

- [ALV1997] Alvares, L. O., Sichman, J. S. *Introdução aos Sistemas Multi-Agentes*. In: Medeiros, Cláudia Maria Bauzer (Ed.). *Jornada de Atualização em Informática (JAI'97)*. Brasília: UnB, 1997. cap. 1, p. 1–38.
- [BRA1988] Bratman M.; Israel D.; Pollack, M. *Plans and Resource-Bounded Practical Reasoning*. *Computational Intelligence*. Stanford, Califórnia, 1988. 4(4):349-355.
- [BOL2002] Bolzan, W., Giraffa, L. M. M. *Estudo comparativo sobre Sistemas Tutores Inteligentes Multi-Agentes Web*. Relatório Técnico, PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Faculdade de Informática. Jul, 2002.
- [BOR2001] Bordini, R. H., Vieira, R., Moreira, Á. F. *Fundamentos de Sistemas Multi-Agentes*. In: Congresso da Sociedade Brasileira de Computação (SBC2001), XX Jornada de

Atualização em Informática (JAI), 21. Anais. Fortaleza-CE, Brasil: Sociedade Brasileira de Computação, 2001. p 3–41.

[BOR2003] Bordini, R. H. e Vieira, R. *Linguagens de programação orientadas a agentes: uma introdução baseada em AgentSpeak(L)*. Revista de Informática Teórica e Aplicada. Instituto de Informática da UFRGS, Brazil, 2003. X(1):7–38.

[BOR2005] Bordini R. H.; Hübner J. F.; *et al.* *Jason - A Java-based agentSpeak interpreter used with saci for multi-agent distribution over the net*. Department of Computer Science University of Durham Durham DH1 3LE, U.K. 2005.

[CAR1970] CARBONELL, J. R. AI in CAI: *na artificial intelligence approach to computer assisted instruction*. IEEE Transactions on Man Machine Systems. 1970. v.11, n.4, p.190-202.

[COS1996] Costa, E. B. *Um modelo de Ambiente Interativo de Ensino - Aprendizagem baseado numa Arquitetura Multi-Agentes*. Campina Grande: CPGEE/UFPa, 1996.

[FIN1992] Finin, T., Fritzson, R., and McKay, D. *A language and protocol to support intelligent agent interoperability*. In Proceedings of the CE & CALS Washington Conference.1992.

[FIN1994a] Finin, T., Fritzson, R., McKay, D., McEntire, R. *KQML as an agent communication language*. In Proceedings of the Third International Conference on Information and Knowledge Management. ACM Press, 1994a.

[FIN1994b] Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, R., McGuire, J., Shapiro, S., and Beck, C. *Specification of the KQML agent-communication language (plus example agent policies and architectures)*. Draft, The DARPA Knowledge Sharing Initiative - External Interfaces Working Group, 1994b.

[FRA1998] Frasson, C.; Martin, L.; Gouarderes, G.; Aimeur, E. LANCA. *A Distance Learning Architecture Based on Networked cognitive Agents*. In Lectures Notes in Computer Science.

Intelligent Tutoring Systems. Proceedings of 4th International Conference, ITS 1998, San Antonio, Texas, August 1998. P.594-603.

[GEN1994] Genesereth, M. R. and Ketchpel, S. P. *Software Agents. Communications of the ACM*, 1994. 37(7):48–53.

[GEO1987] Georgeff, M. P. e Lansky, A. L. *Reactive reasoning and planning*. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI'87), 13–17 July, 1987, Seattle, WA, pages 677–682, Manlo Park, CA. AAAI Press / MIT Press.

[GIR1999] Giraffa, L. M. M. *Uma arquitetura de tutor utilizando estados mentais*, tese de Doutorado, UFRGS - Universidade Federal do Rio Grande do Sul, Instituto de Informática Programa de Pós-Graduação em Computação, Porto Alegre, Mai 1999, 43-59p.

[HAL1997] Halfpap, D. M., Belli, M. J., *O Papel de Sistemas Inteligentes para o desenvolvimento de Organizações Virtuais no ambiente*, Internet, Universidade Federal de Santa Catarina, 1997. [www.ufsc.br}. Acessado em 10/01/2006.

[HUB2003] Hübner, J. F., Sichman, J. S. *Saci Programming Guide*, USP - Universidade de São Paulo, São Paulo, jul 2003.

[HUB2004] Hübner, J. F., Bordini, R. H., Vieira, R. *Introdução ao desenvolvimento de Sistemas Multiagentes com Jason*. Departamento de Sistemas e Computação, Universidade Regional de Blumenau (FURB). 2004, p. 1-12.

[JEN1998] Jennings, N. R. e Wooldridge, M. J. *Agent Technology: foundations, applications, and markets*. Springer Verlag, London, 1998.

[MOU1996] Moussale, N.M., Viccari, R.M., Corrêa M. *Intelligent Tutoring Systems Modelled Through the Mental States*. In: Simpósio Brasileiro de Inteligência Artificial - SBIA, 13, 1996. Berlin: Springer-Verlag.

[MUR1999] MURRAY, T. *Authoring Intelligent Tutoring Systems: Na analysis of the state of the art*. International Journal of Artificial Intelligence in Education ,1999, 10, pages 98-129.

[OLI1994] Oliveira, F. M. *Cr terios de equil bra  o para Sistemas Tutores Inteligentes*. Porto Alegre: CPGCC da UFRGS, 1994. Tese de Doutorado.

[RAO1992] Rao, A. S. e Georgeff, M. P. *An abstract architecture for rational agents*. In Rich, C., Swartout, W. R., e Nebel, B., editors, Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92), 25–29 October, Cambridge, MA, 1992. pages 439–449, San Mateo, CA. Morgan Kaufmann.

[RUS1995] Russell, S. J., Norvig P. *Artificial Intelligence: A Modern Approach*. Prince Hall, Englewood Cliff, New Jersey, 1995.

[WEI1999] Weiss, G. *Multyagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 1999, p. 121 Ed. Gerhard Weiss.

[WOO2000] Wooldridge, M. J. *Reasoning about Rational Agents*. The MIT Press, Cambridge, MA, 2000.

[WOO1999] Wooldridge, M. J. *Intelligent agents*. In Wei , G., editor, Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence. Cambridge, MA 1999. Chapter 1, pages 27 - 77. MIT Press.

[WOO2005] Wooldridge, M. J. *An Introduction To Multiagent Systems*. Ed: John Wiley e Sons, Ltd, March 2005.

3 O Sistema de Gestão de Materiais e Medicamentos - Sig2m

3.1 Histórico

O Sig2m (“Sistema de Gestão de Materiais e Medicamentos”) é um sistema de gestão de estoques e suprimentos de materiais, construído a partir de um convênio entre a Unicamp e as Prefeituras Municipais de Campinas e Guarulhos, para a informatização de seus almoxarifados. O projeto foi desenvolvido pelo Laboratório de Redes de Comunicação (LaRCom) da Faculdade de Engenharia Elétrica e Computação da Unicamp sob, coordenação do Prof. Dr. Leonardo S. Mendes.

O Sig2m foi projetado com base na análise dos procedimentos de manipulação de materiais e medicamentos no almoxarifado da saúde das Prefeituras Municipais de Campinas e de Guarulhos. O Sig2m foi construído e orientado à sistemática de funcionamento do *almoxarifado central*, ou seja, todas as ações relevantes comumente realizadas no almoxarifado foram mapeadas para o sistema sob a forma de processos. Atualmente, o município de Campinas utiliza o Sig2m para a gestão de seu almoxarifado na área da saúde (medicamentos e materiais afins). Entretanto, em Guarulhos o sistema foi implantado para a gestão de todos os almoxarifados.

3.2 A Arquitetura da Rede Municipal de Saúde – Distribuição de Materiais

3.2.1 Disposição dos Setores

A concepção do projeto foi feita sobre uma realidade encontrada na estrutura de saúde dos municípios, na qual se tem um *almoxarifado central* que recebe medicamentos e outros materiais de diversas fontes fornecedoras e redistribui esses materiais para os diversos postos de saúde do município (Unidades Básicas de Saúde - UBSs), conforme ilustrado na

Figura 11. Uma vez na UBS, os materiais são armazenados em um *almoxarifado local* (farmácia) até que sejam consumidos pelos pacientes.

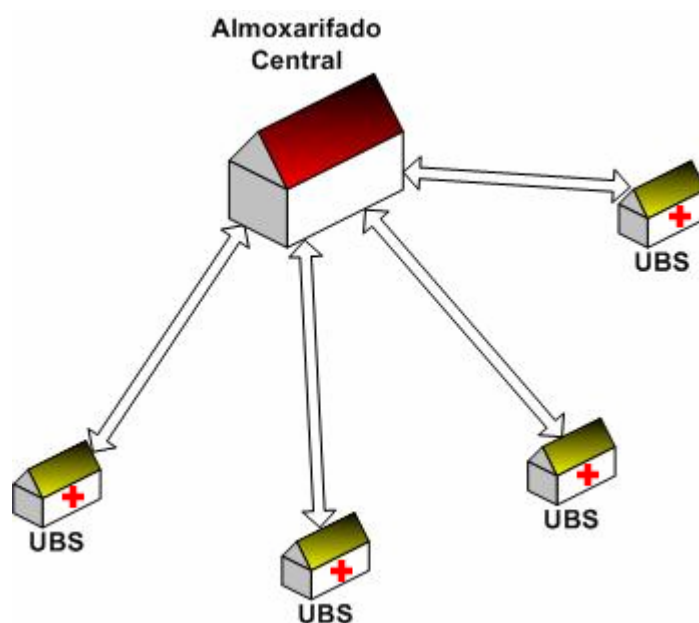


Figura 11 - A arquitetura da rede municipal de saúde.

3.2.2 Disposição de Hardware e Software

O Sig2m foi projetado para suprir inúmeras bases de dados distribuídas a partir de uma base de dados central. A base de dados central gerencia o estoque do almoxarifado central e cada UBS possui uma base de dados local para registrar seu controle local de materiais (conforme ilustrado na Figura 12). O fluxo de comunicação entre o almoxarifado central e as UBSs é controlado através do Sig2m, que dá suporte ao recebimento e envio de requisições de materiais ao almoxarifado.

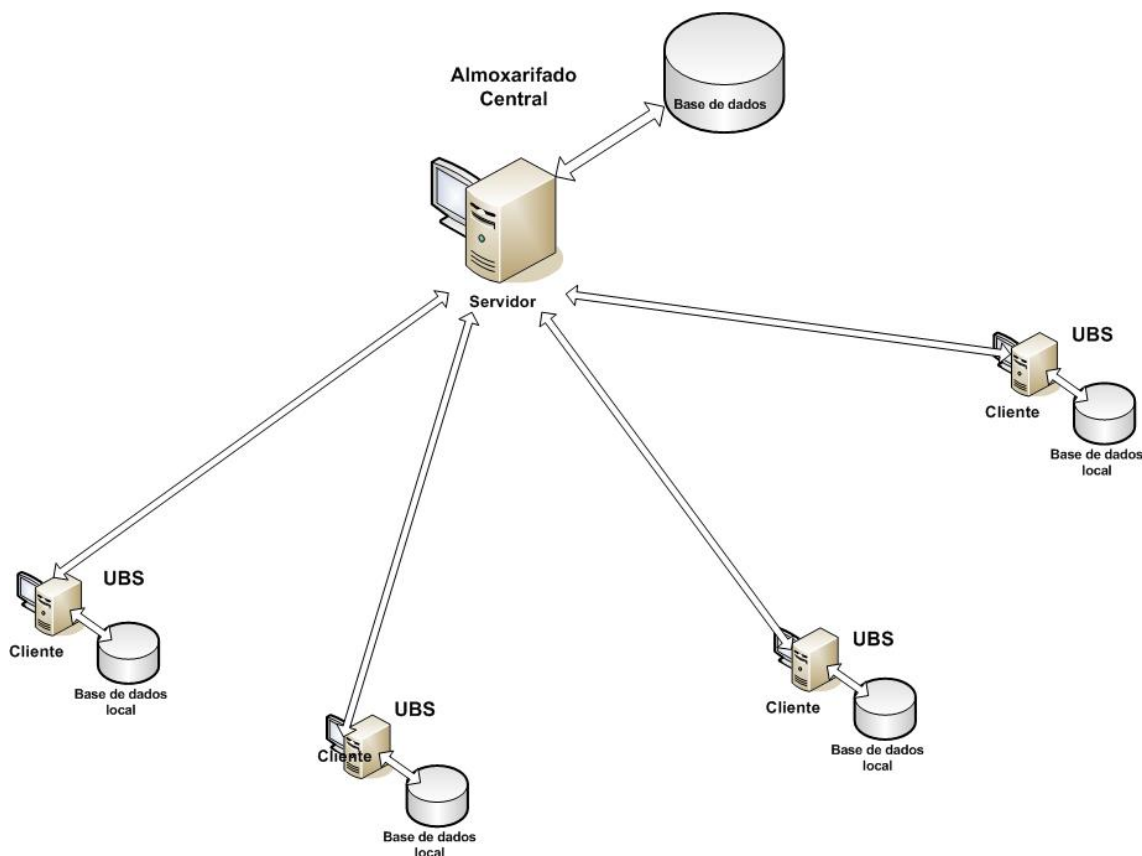


Figura 12 - Estrutura da rede municipal de saúde – visão de software.

3.2.3 Tecnologias Utilizadas para a Construção do Sig2m

O Sig2m é uma aplicação do tipo de *Desktop* (*Java Swing*) construída com a linguagem de programação Java - *Java 2 Standard Edition* (J2SE) versão 1.2, usando um modelo que permite o gerenciamento de processos distribuídos [JAV2007]. As principais características da linguagem Java são:

- Orientação a Objetos - Linguagem de programação orientada a objetos;
- Portabilidade - É uma linguagem multiplataforma, operável em vários sistemas operacionais;
- Multi-tarefa: Possui infra-estrutura para a programação *multi-thread*, que são subprocessos independentes que executam de forma concorrente em um processo-mãe, possibilitando criar programas distribuídos e multi-tarefas;

- Plataforma aberta: A linguagem JAVA é uma arquitetura aberta e extensível, permitindo o desenvolvimento de bibliotecas adicionais (muitas também abertas) o que facilita no desenvolvimento de programas complexos;
- Diversidade de mecanismos de comunicação: Provê suporte para os três tipos de mecanismos de comunicação inter-processos: *Sockets*, RMI e CORBA;
- Segurança: Pode executar programas via rede com restrições de acesso e execução;

O sistema gerenciador de banco de dados (SGBD) utilizado para a implementação da camada de persistência⁴ do Sig2m foi o MySQL 3.X, é um software livre que utiliza a linguagem padrão *Structured Query Language* (SQL) - Linguagem de Consulta Estruturada, como interface. O MySQL baseia-se em princípios de código aberto até a versão 3.X, e suporta várias plataformas: Windows, Linux, FreeBSD, BSDI, Solaris, Mac OS X, SunOS e SGI [MYS2007].

3.2.4 Arquitetura Sistêmica

O Sig2m foi construído baseado em uma arquitetura computacional cliente-servidor. Em uma arquitetura cliente – servidor (como o próprio nome sugere), o sistema é composto de um ou mais elementos servidores e vários elementos clientes. Os clientes solicitam serviços que são fornecidos pelos servidores, conforme ilustrado na Figura 13. Os servidores são responsáveis por prover serviços aos clientes através de uma ou mais interfaces de acesso. Os clientes utilizam estes serviços através de operações de invocação a estas interfaces [BAS2003] [SOM2003].

⁴ Camada de persistência: camada responsável por gerenciar o armazenamento dos dados em um banco de dados, [WIK2007].

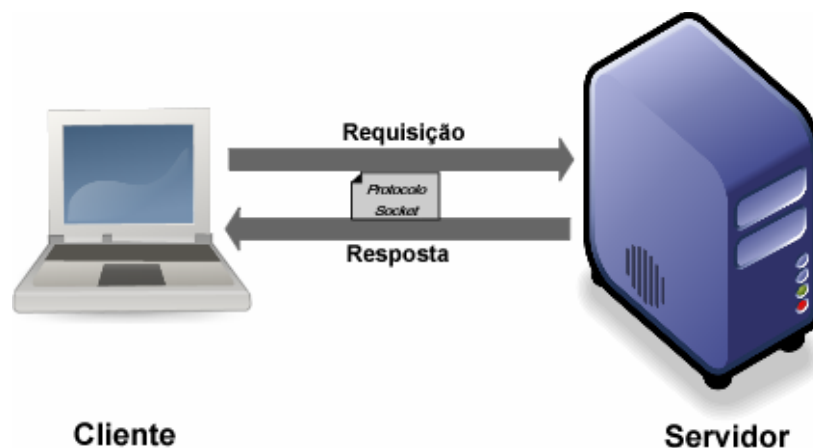


Figura 13 - Arquitetura cliente – servidor.

A relação entre o cliente e servidor é estabelecida apenas durante a execução do serviço. Terminada a execução do serviço, o cliente se desconecta do servidor e continua o seu processamento local. A arquitetura cliente-servidor deve seguir um protocolo de interação para a descrição das interfaces de serviços e para a invocação de tais serviços pelo cliente.

No Sig2m a aplicação cliente (*Desktop*) possui acesso remoto à base de dados central, situada no *Almoxarifado da Saúde* (Almoxarifado da Central), através do sistema servidor. Este acesso é feito via *Socket*⁵, e os clientes estão distribuídos na rede de forma que cada UBS possui uma estação cliente com a sua base de dados local. Levando em consideração a arquitetura citada, o Sig2m possui três versões de instalação:

- Sig2m *Master*: instalação básica destinada ao almoxarifado central, que possibilita a manipulação dos dados centralizados, que tratam do recebimento direto de materiais do fornecedor, o endereçamento destes no almoxarifado central, manipulação e a expedição de materiais para os almoxarifados subordinados.
- Sig2m *Client*: instalação básica destinada às UBS (almoxarifado local), que possibilita toda a manipulação dos materiais, envolvendo o recebimento direto do almoxarifado central ou de fornecedor externo. É responsável pela troca de informações pela rede (Internet), com o almoxarifado centralizador.
- Sig2m *Server*: versão de instalação que provê serviços de acesso dos clientes (UBSs) aos dados contidos na base de dados do almoxarifado central.

⁵ *Sockets*: São interfaces de um canal de comunicação, que permite realizar troca de mensagens entre dois processos (sistemas computacionais) [CAR2007].

O protocolo de troca de mensagens entre o cliente e o servidor é baseado em um modelo de passagem de mensagens, diretamente por *sockets*, conforme ilustra a Figura 14. Neste caso, os dois processos comunicantes, trocam informações entre si diretamente utilizando uma porta de transporte (TCP ou UDP) e mais um endereço de rede, tornando possível identificar um aplicativo ou servidor na rede de comunicação IP. No *socket*, as mensagens podem ser síncronas ou assíncronas. O comportamento global do sistema é assíncrono.

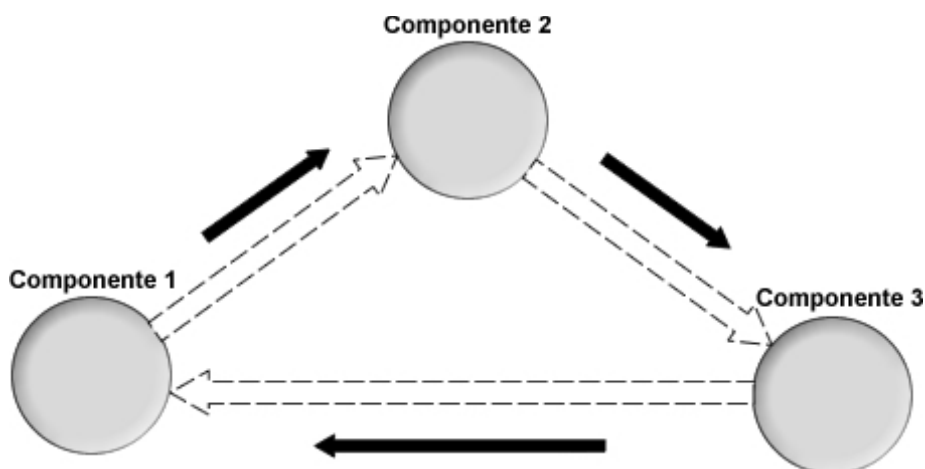


Figura 14 - Modelo de passagem de mensagens - Sockets.

3.3 Principais Limitações do Sistema

3.3.1 Arquitetura cliente-servidor

Uma das principais limitações do sistema está associada aos problemas de escalabilidade e complexidade de implementação. Quando uma considerável quantidade de clientes acessa de forma concorrente o servidor, é gerado um congestionamento de acesso, resultando na perda da entrega das requisições ao servidor, pois o Sig2m não possui um componente de controle que possibilite o aumento da escalabilidade de uso.

A complexidade da troca de informações é outro fator agravante. O acesso concorrente de clientes a um servidor exige um complexo controle de fluxo de mensagens, tais como controle de transação e de segurança, na entrega das informações por parte do cliente.

3.3.2 Módulo de Planejamento de Compra e Gestão de Estoque

O Sig2m ainda não possui um módulo de planejamento de compra de materiais e gestão que faça o apontamento sob o ponto de vista de consumo de itens e sobre o estado do estoque. A falta de tais recursos é um dos motivos da escassez de materiais nos setores e no almoxarifado central. A estimativa das cotas de itens para cada setor ainda é feita de forma manual, forma esta que não garante a sua veracidade, e os valores calculados são enviados para o almoxarifado central através de formulários, que podem ocasionar atrasos.

O Planejamento de Compra e Gestão de Estoque

Gestão de Estoque é o conjunto de atividades e ações que têm por função gerar o pleno atendimento das necessidades da empresa com o máximo de eficiência possível ao menor custo, através do maior giro possível de capital investido. A área de gestão possui técnicas de tratamento que objetivam buscar um equilíbrio entre o estoque e o consumo de um órgão, intermediado por variáveis de suprimentos que acompanham dinamicamente a evolução do estoque. Com os procedimentos de gestão de estoque, torna-se possível realizar o planejamento de compra de um estoque, estimar quando e quanto adquirir de materiais para repor o estoque, gradativamente repercutindo em um estoque próximo ao ideal.

3.3.3 Facilidade de Uso

Durante o processo de implantação do Sig2m nos municípios de Campinas e Guarulhos, foram realizadas cursos de treinamento para os usuários das UBSs e dos Almoxarifados da Saúde, visando conscientizá-los do novo paradigma de trabalho. Apesar dos esforços realizados pela equipe de treinamento, leva-se um certo tempo para que os usuários utilizem

o sistema de forma satisfatória, além de ocorrer a constante rotatividade das funções dos usuários nos setores. Estes motivos levam à utilização incorreta do sistema, trazendo sérios danos à nova dinâmica de trabalho. Um exemplo a ser citado é a atualização das informações do cliente no servidor, que é feita pelo usuário da UBS.

Outra dificuldade encontrada na rede municipal de saúde é a inexistência de mão-de-obra especializada para realizar trabalhos específicos, como por exemplo, gestores de materiais responsáveis por gerir e tomar decisões sobre a situação do estoque distribuído nas UBSs. Estes profissionais experientes no ramo da logística são conhecedores sobre alguns fatores implícitos que influem nos consumos e, através de tais fatores, realizam a sua previsão de demanda de materiais.

Estes fatos apresentados anteriormente foram comprovados através de um estudo sobre a estrutura da rede municipal de saúde de Campinas, que é composta por setenta e duas UBSs, onde cada setor contém um estoque de materiais de 7000 itens, supridos mensalmente pelo almoxarifado central. Alguns setores chegam a movimentar cerca de 250 itens por dia, desta forma, existe a probabilidade destes itens estarem sendo movimentados de forma errada, ou seja, itens sendo solicitados pelos setores quando o mesmo está em abundância no estoque e não requisitados os itens que estão em falta, repercutindo na má distribuição dos materiais na rede.

3.4 Soluções propostas

Levando em consideração as limitações apresentadas, são sugeridas algumas propostas para o aperfeiçoamento do sistema com vistas à melhoria das deficiências apontadas.

3.4.1 Limitações Arquiteturais

Para a correção das limitações apresentadas pela arquitetura cliente-servidor, propõe-se a utilização de outros padrões arquiteturais para a reconstrução do Sig2m, como *n-Tier* e a Orientada a Serviço [KEE2004], [TID2001].

3.4.1.1 Arquitetura *n-Tier*

É uma variante da arquitetura cliente-servidor, na qual os elementos do sistema distribuído estão organizados em camadas verticais (*tiers*). A arquitetura *n-Tier* comumente utilizada é a arquitetura *3-Tier*, que é composta por 3 camadas, sendo cada uma responsável por um tipo de tarefa:

- **Apresentação:** contém os elementos responsáveis pela interação direta com o usuário;
- **Lógica da Aplicação** (ou lógica de negócio): contém os elementos responsáveis pela lógica da aplicação, faz o processamento da informação;
- **Dados** (*Back-End*): contém os elementos responsáveis pela persistência e consistência dos dados.

Os elementos de uma camada interagem apenas com elementos de camadas adjacentes. Os pontos de interação entre camadas devem explicitar protocolos para descrição de interfaces de serviços e para invocação de serviços disponíveis em cada camada. Por exemplo, o ponto de interação entre as camadas de Apresentação e Lógica da Aplicação utiliza protocolos RMI ou HTTP para a troca de informações, conforme mostra a Figura 15.

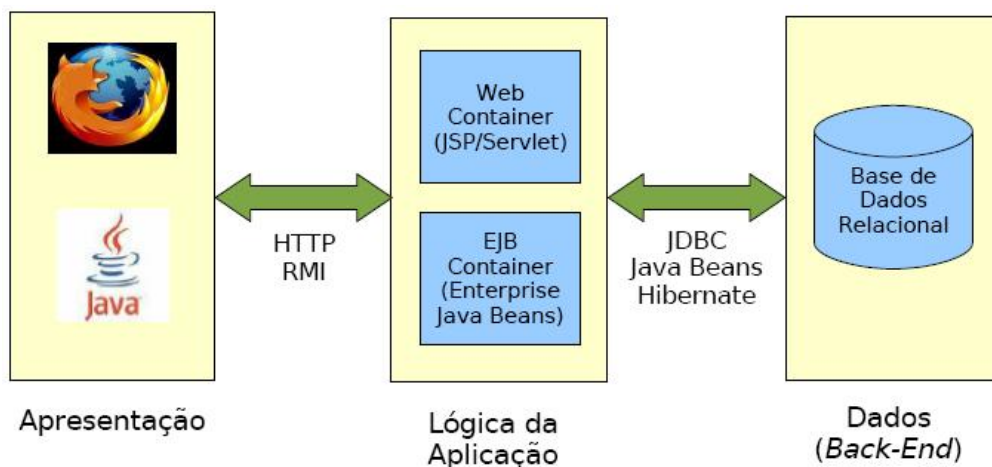


Figura 15 - Arquitetura 3 – Tier, (Extraído de [CAR2007]).

Geralmente, a camada de Lógica da Aplicação fica armazenada em um Servidor de Aplicação. Um Servidor de Aplicação (também conhecido como software de *middleware*), trata-se de um software que é responsável por realizar a administração dos componentes da aplicação. A administração dos componentes envolve o fornecimento dos componentes em tempo de execução, gerência de sessão de acesso, monitoramento, segurança e balanceamento de carga, possibilitando ao desenvolvedor concentrar-se apenas na implementação da camada lógica da aplicação.

O Modelo MVC - Modelo, Visão e Controlador

Um dos modelos mais conhecidos da arquitetura 3-*tier* é o modelo MVC - Modelo, Visão e Controlador (tradução de *Model*, *View* e *Controller*), apresentado pela Figura 16. Esta arquitetura consiste em separar a aplicação em três camadas interoperáveis chamadas Modelo, Visão e Controlador [BUS1996].

A camada “*modelo*” contém o núcleo funcional da aplicação. Ele encapsula dados apropriados e exporta procedimentos que realizam o processamento específico da aplicação (regras de negócio). Os componentes controladores, por sua vez, invocam estes procedimentos, de acordo com as ações do usuário. O modelo também provê funções de acesso a dados que são utilizados pelo componente *visão* para apresentação do mesmo. Um modelo pode ter várias visões associadas a ele.

A camada “*visão*”, transforma o modelo em uma forma apropriada para interação (normalmente uma interface de usuário), que apresenta ao usuário, dados provenientes de um processamento realizado pelo modelo. Diferentes visões apresentam informações do modelo em diferentes formas. Cada visão possui um único controlador que, por sua vez, é associado ao modelo.

O componente “*controlador*” define o comportamento da aplicação, responde às ações do usuário (eventos) e provoca mudanças nos componentes modelo e visão. Ele interpreta os eventos do usuário, tais como entrada de teclado e cliques de *mouse*, e as mapeia para as chamadas do modelo. Os eventos são traduzidos em requisições de serviços ao modelo, o qual, realiza ativação das regras de negócio efetuando algum processamento. Com base nas ações do usuário (gerada na visão) e nos resultados do processamento no modelo, o

controlador seleciona uma visão a ser exibida. Em geral, existe um controlador para cada conjunto de funcionalidades relacionadas do sistema.

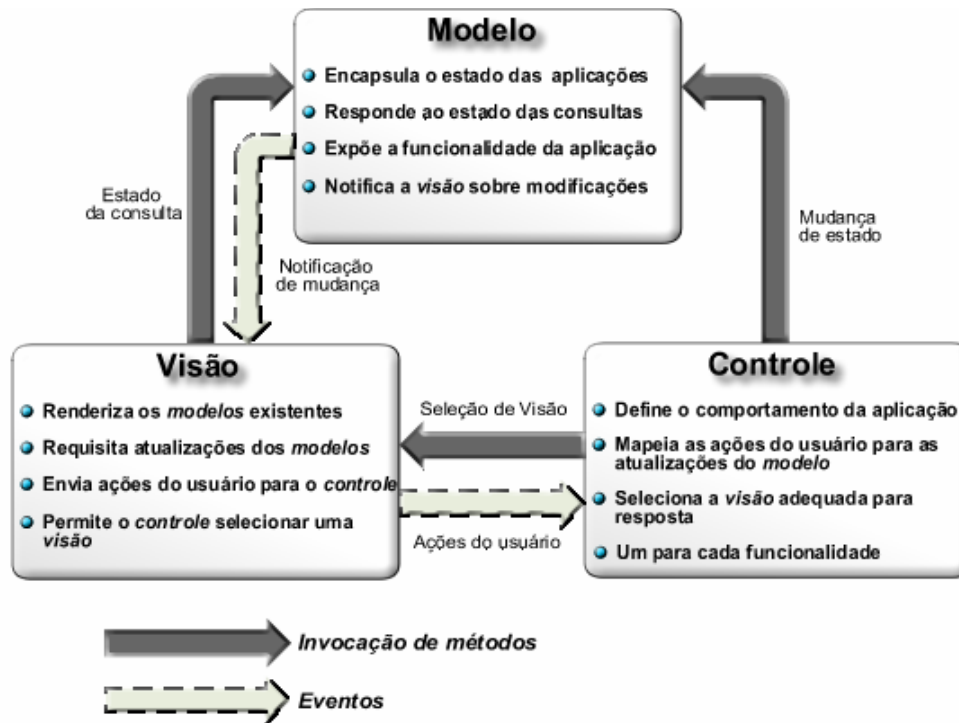


Figura 16 - Arquitetura MVC – Model, View e Controller.

3.4.1.2 Implementação do Sig2m sob a Arquitetura 3 - Tier

O Sig2m poderia ser reestruturado em três camadas, conforme a arquitetura 3-tier. O fato de uma estação Sig2m estar dividida em camadas facilitaria o acesso de uma estação aos recursos disponíveis de outra, como por exemplo, um componente da camada de Lógica de Aplicação da estação A acessar a camada de dados da estação B (vide Figura 17). Todo o gerenciamento de acesso entre o uso dos componentes seria regido pelo Servidor de Aplicação. Assim, problemas de acesso às informações de nível cliente-cliente e de congestionamento no servidor seriam sanados.

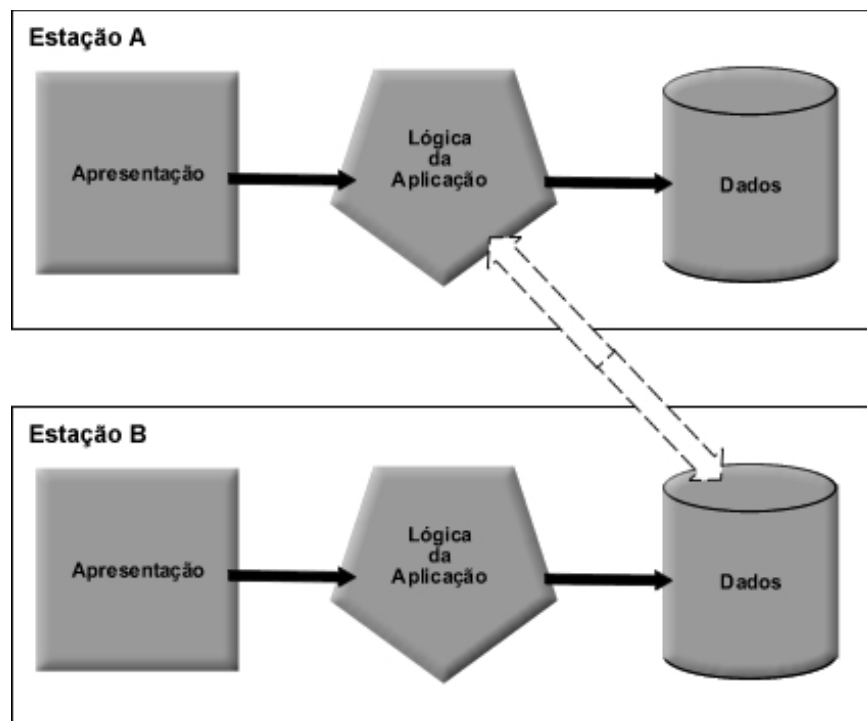


Figura 17 - Implementação do Sig2m sob arquitetura 3 -Tier.

3.4.1.3 Arquitetura Orientada a Serviço

A Arquitetura Orientada a Serviço (*Service Oriented Architecture* - SOA) é um tipo de arquitetura de processamento distribuído inter-organizacional, que tem como propósito manter a interoperabilidade e a segurança da informação utilizando uma arquitetura baseado no conceito de serviços. Serviços são aplicativos ou rotinas que são disponibilizadas em uma rede de computadores que encapsulam uma lógica de negócio, dotadas de uma interface de acesso independente de outros serviços. No SOA, os serviços podem ser acessados sem a necessidade de conhecer ou entender a sua plataforma de implementação.

Atualmente o SOA vem sendo implementado, utilizando a tecnologia baseada em serviços web (*Web Services*). Pode-se dizer que SOA baseado em *Web Services* é o corrente estado da arte na integração entre sistemas, pois esta tecnologia atende aos requisitos de interoperabilidade, através do uso de protocolos como HTTP, XML e SMTP, que são padrões de tráfego de informação para a *web* baseados no formato de texto [TID2001] [KEE2004].

A arquitetura de “*Web Service*” funciona como interface posicionada entre código da aplicação e o usuário provendo uma camada de abstração entre este e o código da aplicação, conforme ilustrado na Figura 18.

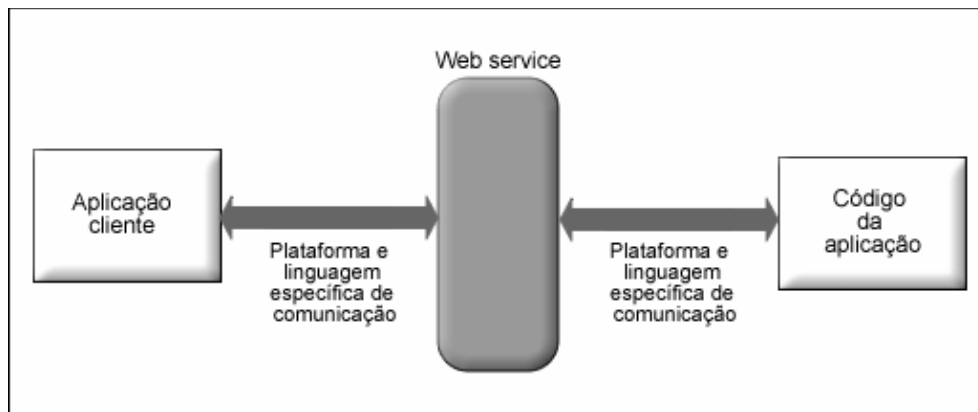


Figura 18 - Funcionamento da arquitetura de web services (Extraído de [TID2001]).

A estrutura de *Web Services* permite dividir uma aplicação em três camadas:

- Camada de Aplicação: plataforma de suporte a serviços;
- Camada de Negócio: detém as regras de negócio definidas e suas políticas de uso;
- Camada de Serviço: representa as interfaces e conexões entre os serviços, que são elementos de ligação entre a camada de aplicação e a de negócio.

Na arquitetura de *Web Services*, o Provedor do Serviço descreve e publica o(s) serviço(s), sob forma de interface de invocação em um Registro de Serviços. As interfaces de invocação dos serviços são escritas em um arquivo descritor que possui formato *eXtensible Markup Language* (XML), conhecido como *Web Services Description Language* (WSDL). Quando o Consumidor do Serviço quer fazer uso de um serviço, este deve pesquisar e recuperar o serviço que necessita no Registro de Serviços, utilizando um protocolo chamado *Universal Description, Discovery and Integration* (UDDI). O Registro de Serviços retorna um documento conhecido como *Web Services Description Language* (WSDL) que contém especificações de localização do serviço, possibilitando o seu uso pelo consumidor. Este processo pode ser visualizado pela Figura 19. O tráfego da informação entre o serviço e seu requisitante é realizado por um protocolo chamado *Simple Object*

Access Protocol (SOAP). O protocolo SOAP é um conjunto de regras para representar a informação em formato XML.

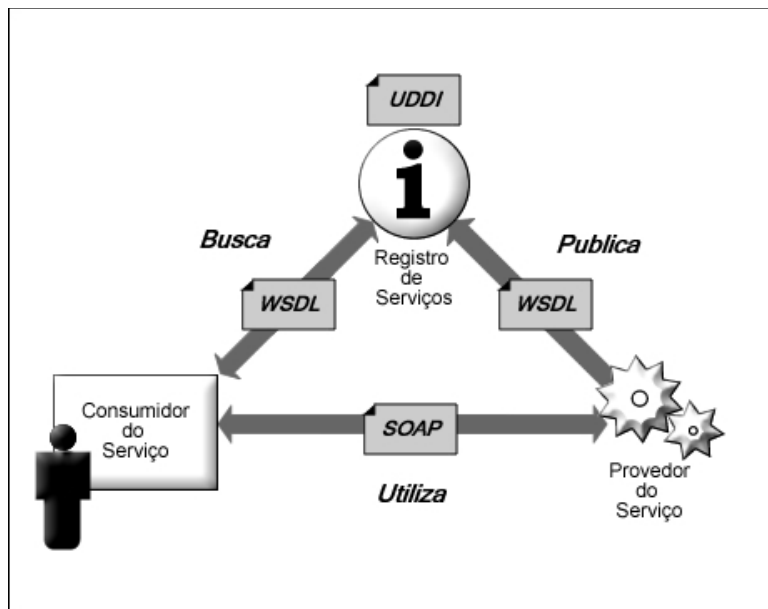


Figura 19 - Arquitetura de SOA - Web Services, adaptado de [TID2001].

As principais características da Arquitetura Orientada a Serviços são [BOO2007] [BIE2007]:

- Baixo acoplamento entre aplicações;
- Alta interoperabilidade entre plataformas tecnológicas;
- Alta reutilização dos serviços (regras de negócio);
- Resposta rápida a mudanças nos processos de negócio;
- Serviços são interoperáveis;
- Transparência sobre a implementação dos serviços;
- Serviços disponibilizados em plataforma Web;
- Interfaces especificadas em XML na sintaxe WSDL;
- Mensagens trocadas através de requisições HTTP no protocolo SOAP (via padrão XML);
- Podem ser implementados em diferentes tecnologias como Java, .Net, PL/SQL, C++, entre outros.

3.4.1.4 Implementação do Sig2m sob a Arquitetura de *Web Services*

Em uma proposta de arquitetura de *Web Services* para o Sig2m, os módulos estariam dispostos sob forma de serviços, em que cada estação do Sig2m acessaria os seus próprios serviços e o de outras estações, através de permissões específicas de acesso. Por exemplo, a estação A do Sig2m pode acessar o módulo de relatórios pertinentes à estação B, conforme ilustrado na Figura 20.

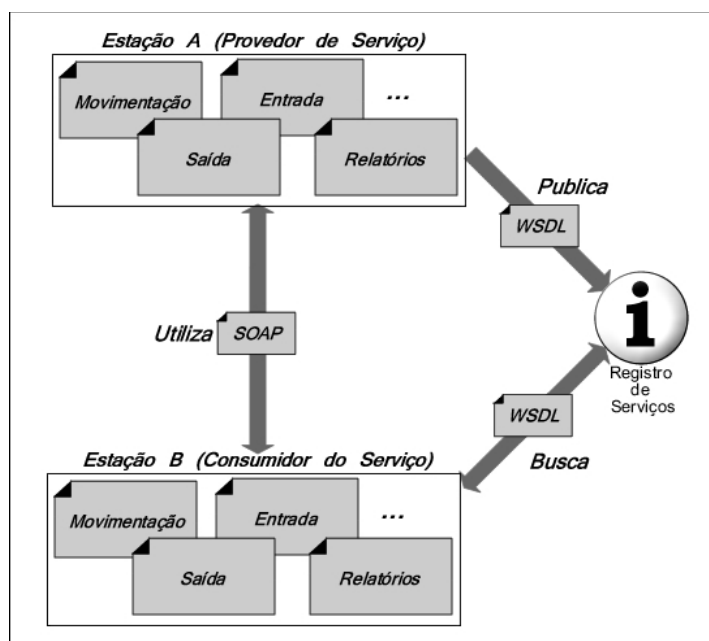


Figura 20 - Implementação do Sig2m sob arquitetura de Web Services.

3.4.2 Soluções propostas para as Limitações de Gestão de Materiais e Recursos Humanos

O principal objetivo deste trabalho é o estudo de tecnologias alternativas para desenvolvimento dos projetos de e-Gov, que vêm sendo produzidos pelo LarCom como fruto dos convênios entre este laboratório e prefeituras municipais, dentro do escopo maior do projeto “*Infovias Municipais*”. Desta forma, nosso estudo se fixa num dos sistemas que foram desenvolvidos dentro destes convênios: O Sig2m. Assim, buscamos alternativas para desenvolver soluções que corrijam os problemas identificados com este sistema.

Os problemas encontrados com o Sig2m sugerem a reestruturação do sistema a partir das duas implementações arquiteturais apresentadas. Entretanto, neste trabalho, iremos nos ater na implementação do módulo de gestão de materiais e de recursos humanos através do uso da sociedade de Sistemas Tutores Inteligentes (STIs) modelados sobre a arquitetura de Sistemas Multi-Agentes. Os STIs são sistemas de natureza educacional que têm por função auxiliar um usuário (em sua aprendizagem) na realização de determinadas tarefas, tais como: alertar quando um apontador do sistema está atingindo um nível de risco; avaliar se o usuário está desempenhando com êxito suas tarefas a partir de um conteúdo dado; apresentar diretivas para o usuário realizar uma determinada tarefa; etc. Atualmente, os STIs estão sendo implementados com o emprego de agentes, na tentativa de produzir sistemas mais inteligentes, flexíveis e independentes. O uso de uma sociedade de STIs sobre um arquitetura do Sig2m será apresentado no decorrer deste trabalho.

3.5 Bibliografia

[BAS2003] Bass L., *et al. Software Architecture in Practice*. 2ª Ed., editora: Addison Wesley. Canadá, 2003.

[BUS1996] Buschmann, F. *et al. Pattern – Oriented, Software Architecture. A System of Patterns*, 1996.127p. Editora: JOHN WILEY & SONS, New York - EUA.

[KEE2004] Keen, M., *et al. Patterns: Implementing an SOA Using an Enterprise Service Bus*, 2004. Editor: redbooks, EUA.

[SOM2003] Sommerville, I. *Engenharia de Software*. 6ª Ed. São Paulo – SP. Tradução André Maurício de Andrade Ribeiro, 2003. Editora: Pearson Addison Wesley.

[TID2001] Tidwell D., Snell J., *et al. Programming Web Services with SOAP*, 2001. Editora: O'Reilly, Canadá.

Links Visitados

[BIE2007] Bieberstein N.; *et al. Service-Oriented Architecture Compass by Pearson*, 2006. Disponível em: <http://pt.wikipedia.org/wiki/SOA#Refer.C3.AAncias>, acessado em 07/04/2007.

[BOO2007] Booth, D. *Web Services Architecture*, 2004. Disponível em: <http://www.w3.org/TR/ws-arch/#technology>, acessado em 07/04/2007.

[CAR2007] Cardozo, E., IA-847 - *Projeto de Sistemas Distribuídos Abertos*, disponível em: <http://www.dca.fee.unicamp.br/~elери/posgrad.html>, acessado em 07/04/2007.

[JAV2007] Java Technology, disponível em: <http://java.sun.com>, acessado em 03/04/2007.

[MYS2007] MySql, disponível em: <http://www.mysql.com/>, acessado em 03/04/2007.

[WIK2007] Wikipedia, disponível em: <http://pt.wikipedia.org/>, acessado em 03/04/2007.

4 Abordagem de Sociedades de STI como instrumento de auxílio à gestão de materiais e medicamentos

4.1 Introdução

Este capítulo aborda a construção de um STI, arquitetura de sistema com suporte a auxílio à aprendizagem, orientado a operar na área de administração de materiais, mais especificamente gestão de materiais e medicamentos na rede municipal de saúde de um município. O STI proposto foi construído com o intuito de ser uma ferramenta de assistência e suporte para tomada de decisões, instruindo o usuário de cada Unidade Básica de Saúde (UBS) a tomar as providências necessárias para obter um estoque padrão de materiais e medicamentos próximos as suas reais necessidades.

Para uma melhor compreensão deste trabalho serão inicialmente apresentados o funcionamento do sistema de distribuição de materiais e medicamentos da rede municipal de saúde e a contextualização dos problemas com os quais este se depara. Em seguida, será mostrada a arquitetura geral da Sociedade de Tutores Inteligentes e, em seqüência, enfatizados os detalhes de cada componente de sua estrutura.

4.2 Arquitetura da Rede Municipal de Saúde

4.2.1 O processo de distribuição de materiais

O processo de distribuição de medicamentos que foi adotado como objeto desta pesquisa, utiliza a arquitetura apresentada na Figura 21. Neste caso, um almoxarifado gestor central é responsável por suprir todas as UBSs da rede, conforme Mattos [MAT2006].

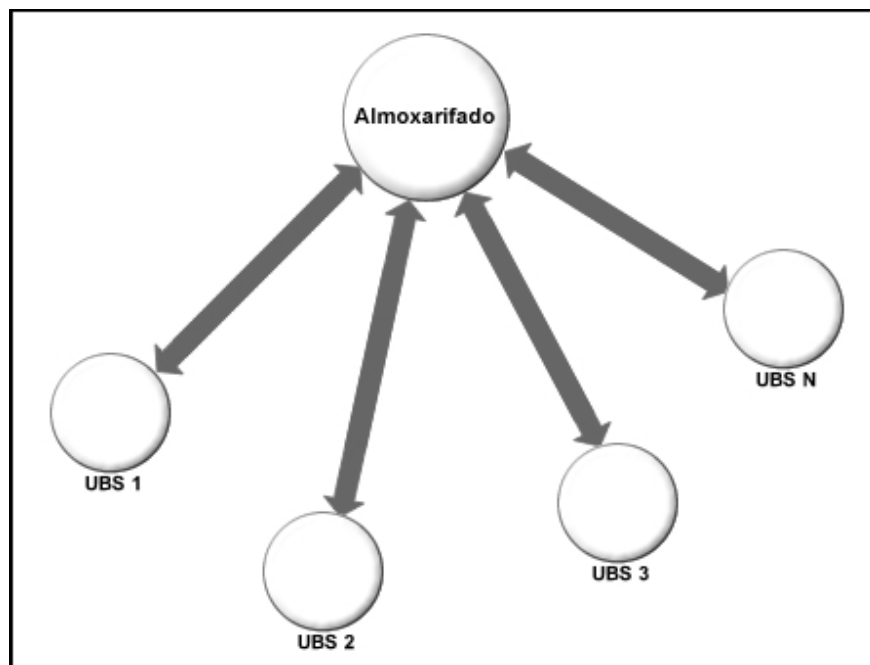


Figura 21 - Arquitetura da rede de Municipal de Saúde (UBSs e Almojarifado Gestor).

De forma regular e sincronizada, as UBSs são supridas com medicamentos de uma lista padrão de medicamentos, obedecendo uma cota que varia entre as UBSs. O ciclo de distribuição é mensal para que sempre haja uma melhor rotatividade do estoque. Caso haja a falta de um medicamento de certa relevância, ou seja, do tipo Z, requisita-se o suprimento imediato do almojarifado, que o fará se o possuir em estoque.

Na véspera do término de cada ciclo, o usuário gestor responsável de cada UBS faz o processo de estimação de cota dos itens para o mês seguinte. O recálculo das cotas de cada setor é enviado para o almojarifado gestor, onde são analisadas e, posteriormente, atendidas.

O sistema de controle de estoque adotado neste trabalho, e que contempla a arquitetura comentada na Figura 21 é o Sig2m. O Sig2m é um software para gerenciamento de almojarifado, desenvolvido pelo LaRCom/FEEC/Unicamp⁶ para atender almojarifados

⁶ LaRCom/FEEC/Unicamp – Laboratório de Redes de Comunicações da FEEC - Faculdade de Engenharia Elétrica e Computação. Unicamp - Universidade Estadual de Campinas.

fisicamente distribuídos. Sua concepção foi desenvolvida a partir da realidade encontrada na estrutura organizacional dos centros de distribuição de medicamentos para postos de saúde, onde um almoxarifado central gerencia a compra, armazenamento e distribuição de medicamentos e suprimentos para um conjunto de pequenos almoxarifados alocados em UBSs espalhadas por uma ampla região, conforme apresentado no capítulo 3.

4.3 Modelagem das sociedades (Solução Proposta)

Como proposta para a resolução da questão da má distribuição de medicamentos e o monitoramento do estoque, enfatizado no capítulo 1, foi montada uma estrutura de Sociedade de Tutores Inteligentes, a qual tem como objetivo auxiliar o usuário de cada UBS a tratar seu estoque de maneira a otimizá-lo. Esta otimização significa trazer o estoque mais próximo às necessidades das UBSs.

A solução proposta foi construída baseada em duas visões do estoque:

- Estoque local: Consiste na visão sobre o estoque limitado a cada UBS.
- Estoque distribuído: Visão em relação ao estoque das sociedades de UBSs da rede municipal de saúde.

Para realizar o tratamento sobre tais pontos de vista, foram construídas duas sociedades de agentes, denominadas *sociedade Local e Remota*.

4.3.1 Sociedade Local

Cada UBS contém uma sociedade de agentes híbrida formada por um *Agente Gestor Tutor* (agente cognitivo), modelado sob a arquitetura de estados mentais (BDI), e um *Agente Mensageiro* (agente reativo), denominado *Sociedade Local*.

A função da Sociedade Local é auxiliar o usuário a monitorar o estado do estoque dos medicamentos. Esta função consiste em prever, através de cálculos de gestão de estoque, os medicamentos que estão excedentes (para posteriormente serem doados a outras UBSs) e / ou os que irão atingir a margem de risco de esgotamento (para serem requisitados às demais UBSs), antes do tempo de reposição (TR). É também função da Sociedade Local

calcular a *cota ideal* (CI) da lista de suprimento para o ciclo seguinte (detalhes sobre os elementos de gestão de estoque como TR, CI, ponto de pedido (PP) entre outros, serão encontrados no Anexo B deste trabalho).

O usuário é informado sobre o estado do estoque através de mensagens enviadas pelo Agente Gestor Tutor (AGT) intercedido pelo Agente Mensageiro, responsável por prover o canal de comunicação. O agente mensageiro obtém a percepção de uma mensagem enviada pelo AGT e a encaminha ao sistema de controle de estoque, conforme a Figura 22. O sistema de controle de estoque que serve de interface com o usuário e o AGT é o Sig2m, apresentado no capítulo 2.

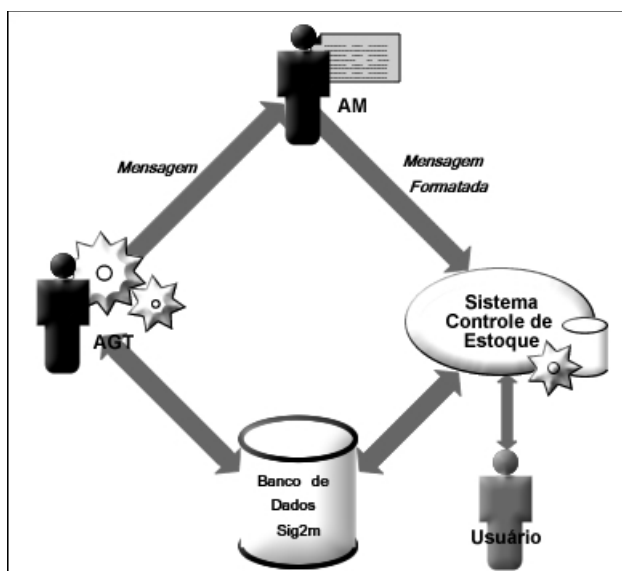


Figura 22 - Arquitetura da sociedade local.

4.3.2 Sociedade Remota

O AGT faz parte de uma segunda sociedade, chamada de *Sociedade Remota* (ou *Sociedade de Tutores Gestores*). A *Sociedade Remota* consiste em uma sociedade de agentes gestores tutores, sendo que cada AGT é representante de uma UBS. A Figura 23 demonstra esta disposição.

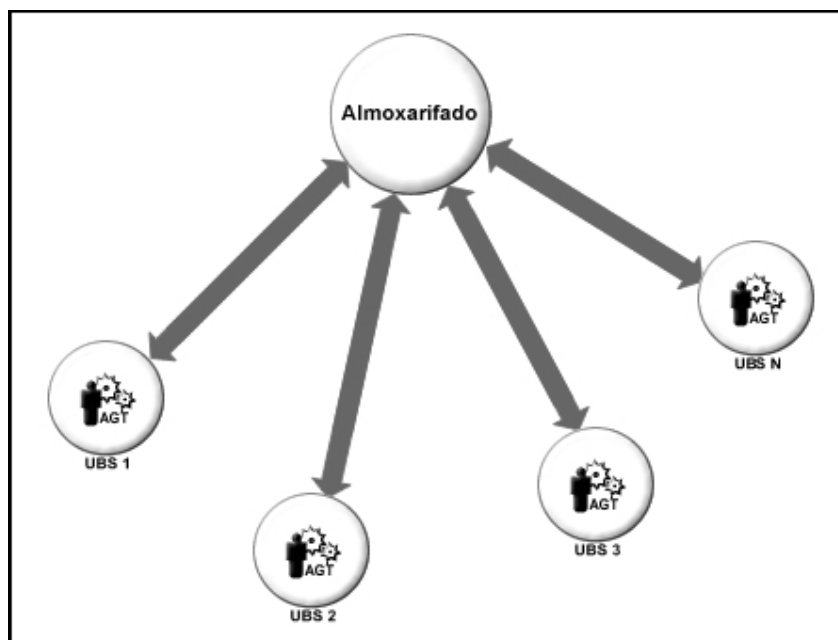


Figura 23 -Disposição da Sociedade Remota sobre a Rede Municipal de Saúde.

A *Sociedade Remota* tem por finalidade gerar a comunicação entre os demais agentes cognitivos da sociedade para realizar o processo de requisição e / ou doação de medicamentos, a partir de uma lista de medicamentos zerados e excedentes de cada UBS, calculada pelo AGT, conforme mostrado na Figura 24. Antes do fim de cada ciclo, o agente gestor envia a cota ideal de cada medicamento para o almoxarifado gestor.

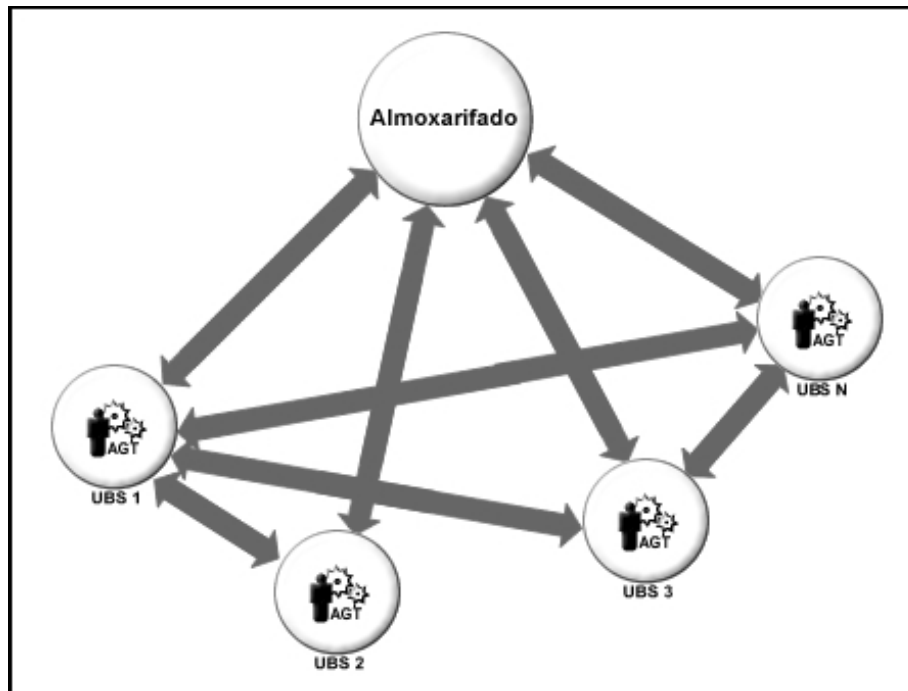


Figura 24 - Processo de requisição e doação de materiais entre UBSs.

4.3.3 Arquitetura das sociedades

A arquitetura das sociedades é moldada sob a arquitetura de comunicação orientada a agentes facilitadores, agente complexo que organiza o trabalho entre os agentes da sociedade. O facilitador é responsável por armazenar informações relevantes como a identidade, localização (endereço na rede) e serviços providos pelos agentes da sociedade [WEI1999]. Toda vez que um determinado agente entra na sociedade, anuncia ao agente facilitador a sua entrada e publica as suas funcionalidades (passos 1 e 2 da Figura 25). Toda a comunicação entre os agentes é intermediada pelo agente facilitador. Por exemplo, quando o agente A pretende se comunicar com o agente B e não possui o seu endereço, ele recorre ao agente facilitador (passo 3) para solicitar o endereço do agente B na rede (passo 4) e então realizar a comunicação (passo 5).

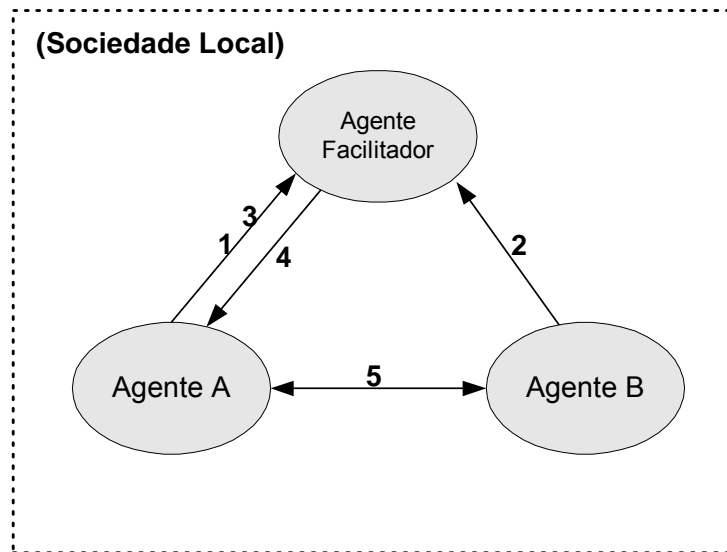


Figura 25 - Sociedade intermediada pelo agente facilitador.

4.3.4 Protocolo de comunicação entre agentes

A comunicação entre os agentes é assíncrona e é feita através de um protocolo de comunicação. O protocolo de comunicação adotado é o *Knowledge Query and Manipulation Language* (KQML), que dá suporte de alto nível à comunicação entre agentes para realizar a troca de conhecimento e informações, [ALV1997] [GEN1994] [WEI1999].

4.4 Funcionamento do Sistema

Em cada ciclo de distribuição de materiais, o AGT de cada UBS anota a entrada da última lista de itens enviada pelo almoxarifado central (processo 1 da Figura 26) e, a partir dela, realiza cálculos de gestão de estoque (processo 2). Com os cálculos de gestão é possível obter uma previsão de *cota ideal* para o próximo ciclo de suprimento, bem como o *saldo ideal* para cada material (processo 3). A partir do saldo ideal, é obtida a lista de itens que estão em estado de risco, denominada *lista de materiais excedentes e zerados* (processo 4). A lista é constituída por itens que irão sobrar e / ou faltar em um período de suprimento bem como pela quantidade prevista em dias de duração do estoque.

Depois de realizados o cálculo do estado geral do estoque e os cálculos de gestão (processo 5 da Figura 26), o AGT adota a estratégia de informar o usuário sobre a situação do estoque, orientando-o a tomar as decisões corretas para a normalizá-lo (processo 6 da Figura 26). Ao observar o estado do estoque, o usuário pode optar por doar e / ou requisitar medicamentos para outras UBSs ou para o almoxarifado central. Este processo é realizado diariamente pelo AGT.

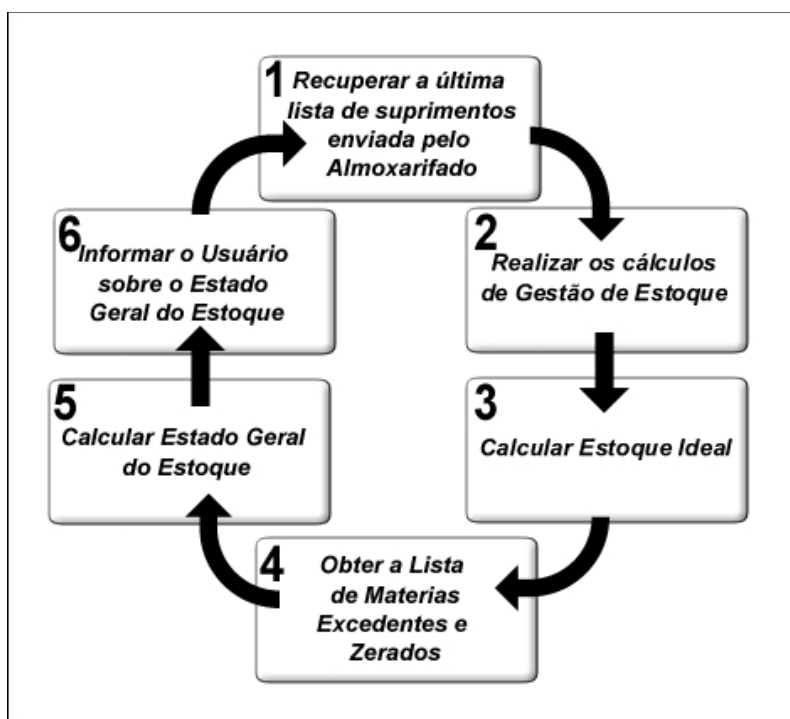


Figura 26 - Ciclo de tarefas do AGT.

Caso a UBS necessite de medicamentos durante um ciclo de distribuição, o usuário gera uma lista de requisição de itens a ser enviada para as demais UBSs. Então, o AGT realiza a ação de encaminhar a lista de requisições para todos os demais agentes gestores da sociedade remota, candidatos à doação, conforme passos 1 e 2 da Figura 27.

O AGT candidato à doação, por sua vez, realiza a percepção da chegada da lista de requisição de medicamentos, armazena-a e realiza a ação de informar ao usuário sobre a sua chegada. Se a UBS, candidata à doação possui itens excedentes, o usuário poderá realizar a doação do material para o setor requisitante. A partir da lista de requisição, o

usuário gera a sua proposta à doação. O AGT doador realiza a ação de enviar para o AGT requisitante a proposta de doação e espera a sua confirmação de aceitação e / ou rejeição (passo 3 da Figura 27).

Ao receber a lista de proposta de doação, o AGT requisitante informa o usuário sobre a sua chegada. O usuário por sua vez, analisa a proposta de doação e confirma a aceitação e / ou rejeição (passo 4 da Figura 27), que é novamente encaminhada para o respectivo setor doador para gerar a sua efetiva saída.

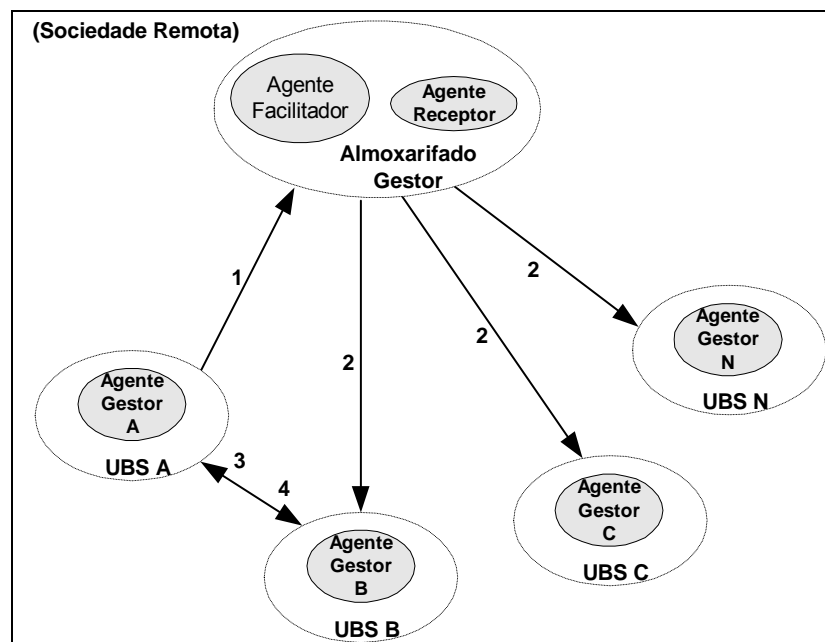


Figura 27 – Detalhes do processo de requisição e doação de medicamentos entre UBSs.

Periodicamente, próximo ao término de cada ciclo, cada AGT da sociedade envia para o almoxarifado gestor a proposta de cota ideal para o suprimento do ciclo seguinte com as cotas corrigidas. No anexo D deste trabalho, é encontrado um exemplo das mensagens KQML trocadas entre 3 AGTs de uma sociedade no processo de requisição e doação de materiais.

4.4.1 Trabalho cooperativo da Sociedade Remota

O trabalho cooperativo da sociedade remota é também conhecido como protocolo de interação entre os agentes da sociedade. Com o intuito de maximizar as utilidades dos agentes, este trabalho consiste em uma série de troca de mensagens, estabelecendo assim, uma conversação de forma estruturada entre os envolvidos. Neste caso, os agentes possuem objetivos similares ou problemas em comum, como é o caso da área da resolução de problemas distribuídos, conforme Huhns [HUH1999].

O objetivo de um protocolo de interação é manter, globalmente, uma coerência de performance de um agente sem violar a sua autonomia. De acordo com Weiss [WEI1999], os importantes aspectos de um trabalho cooperativo incluem:

- Determinar objetivos compartilhados;
- Determinar tarefas comuns a serem realizadas pelos agentes;
- Evitar conflitos desnecessários entre os agentes da sociedade;
- Resolver um problema complexo a partir de sua divisão (subproblemas).

O sistema de coordenação da sociedade remota é baseado no sistema de coordenação distribuída, em que os agentes estão em mesmo nível hierárquico. Desta forma todos os agentes executam e cooperam entre si para resolverem as suas tarefas, conforme apresentado no capítulo 2.

4.5 Arquitetura implementada (Componentes do sistema)

A *Sociedade de Tutores Inteligentes* foi construída sobre três componentes básicos: o *Agente Gestor Tutor* (AGT); o *Agente Mensageiro* (AM); e o *Agente Facilitador*. Nesta seção, serão apresentados em detalhes cada um destes componentes e descritas suas funcionalidades.

4.5.1 Modelagem do Agente Gestor Tutor (AGT)

O AGT foi construído baseado na arquitetura cognitiva, mais especificamente a arquitetura baseada em estados mentais. Esta arquitetura define o estado de um agente, a partir de um conjunto de componentes mentais, tais como: Crenças, Desejos e Intenções (abreviação de BDI, *Beliefs, Desires e Intentions*). Shoham [SHO1993], enfatiza que, para gerenciar a sua autonomia, os agentes possuem inteligência baseada nos próprios estados mentais. Em função dos estados mentais, um agente define o seu próprio comportamento autônomo, que consiste em decidir a ação seguinte a ser executada para atingir um objetivo pretendido.

O conjunto de crenças, desejos e intenções estão codificados sob forma de uma linguagem de programação orientada a agentes, conhecida como *AgentSpeak(L)*. A linguagem *AgentSpeak(L)* foi projetada para a programação de agentes BDI, na forma de sistemas de planejamento reativos (*reactive planning systems*). Sistemas de planejamento reativos são sistemas que estão permanentemente em execução, reagindo a eventos que acontecem no ambiente em que estão situados através da execução de planos que se encontram em uma biblioteca de planos parcialmente instanciados, conforme Hübner [HUB2004].

4.5.1.1 O interpretador Jason

O Jason é um interpretador multi-plataforma para uma extensão de código *AgentSpeak(L)* [BOR2003] e [BOR2005]. Seu código é aberto e sua distribuição é realizada sob a licença GNU LGPL⁷. Ele foi implementado em Java pelo Laboratório de Técnicas Inteligentes da Escola Politécnica da Universidade de São Paulo.

Além de realizar a interpretação do código *AgentSpeak(L)*, o Jason possui outros recursos, como a possibilidade de executar o SMA de forma distribuída em uma rede (usando o *framework* SACI, comentado a seguir) e também de especializar na linguagem Java as funções de seleção de planos, além das principais funções da arquitetura de um agente (percepção, revisão de crenças, comunicação e atuação).

⁷ Disponível em <http://jason.sourceforge.net>, acessado em :10/09/2006

Agentes codificados com o Jason possuem um ciclo de execução constituído por três fases: percepção do ambiente, atualização de crenças e execução das ações, conforme ilustrado na Figura 28. A camada de controle (representada por uma classe Java) faz as percepções do ambiente e as traduzem em um conjunto de crenças. As crenças são enviadas para a camada cognitiva, em que é interpretada pelo interpretador Jason em conjunto com o código *AgentSpeak(L)* do agente. Com o conjunto de crenças, o interpretador seleciona o melhor plano disponível e o coloca em execução, enviando as ações do plano para a camada de controle executar no ambiente. Este ciclo é repetido durante o tempo de vida do agente.

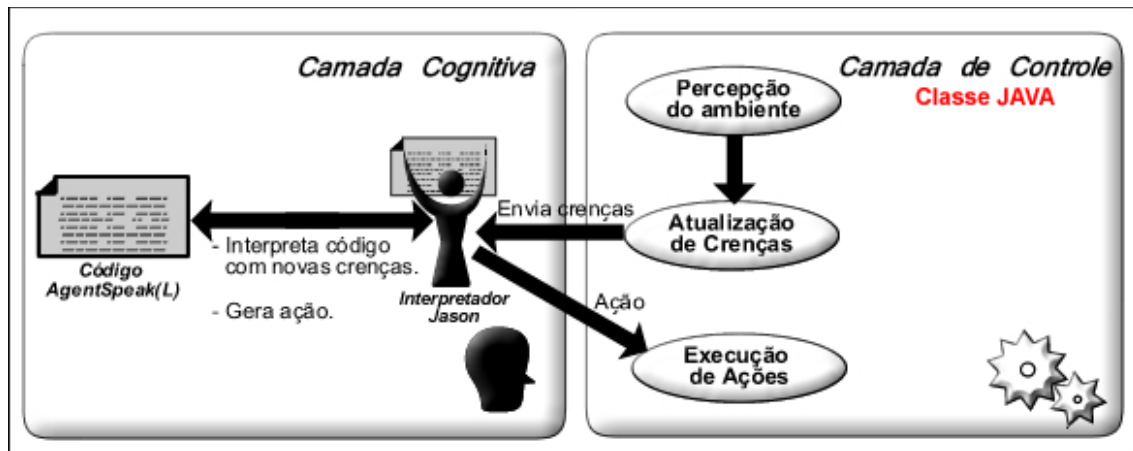


Figura 28 - Funcionamento do Interpretador Jason.

A estrutura interna do AGT é dividida em três camadas interoperáveis: controle; cognitiva; e comunicação, conforme ilustrado na Figura 29. A camada cognitiva é codificada na linguagem *AgentSpeak(L)*. As camadas de controle e comunicação são codificadas na linguagem Java, onde são realizados os processos de percepção, comunicação, revisão de crenças e atuação do AGT. A seguir, será comentado sobre cada camada em questão.

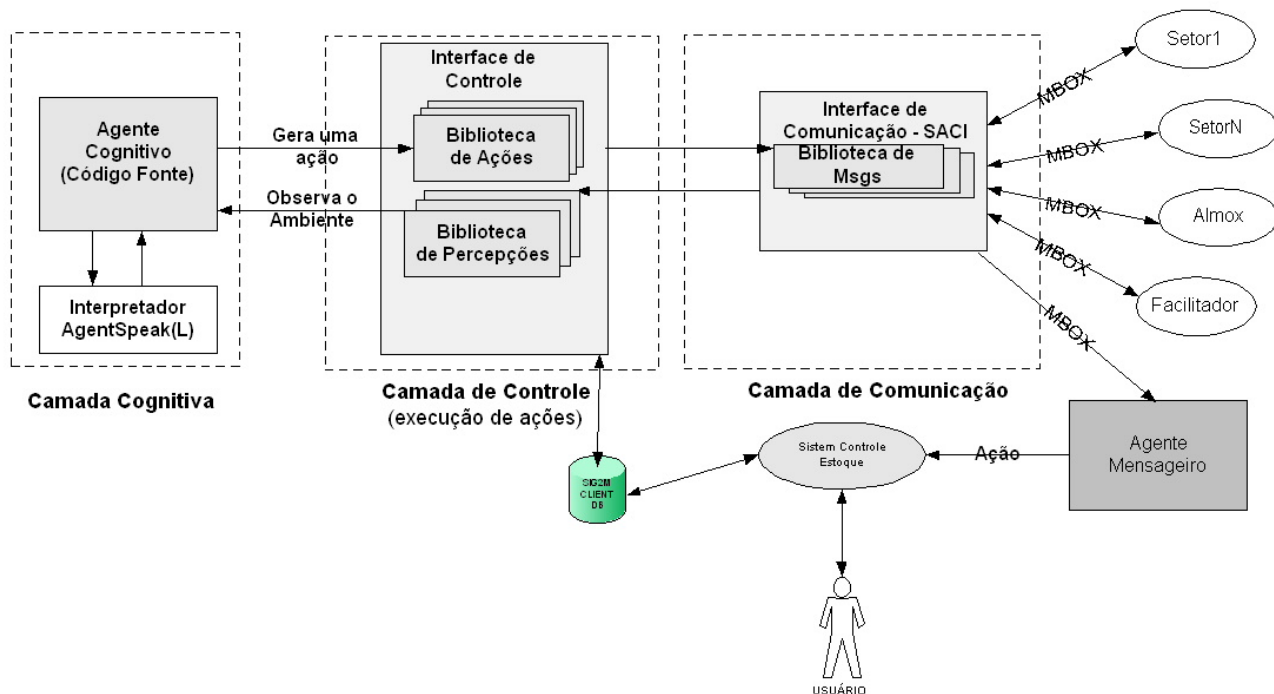


Figura 29 - Anatomia do Agente Gestor Tutor

4.5.1.2 A Camada de Controle

Pode-se dizer que a camada de controle constitui “os olhos e membros”, ou seja, a parte sensorial e atuadora do AGT. É esta que contém os sensores que realizam a percepção do ambiente (crenças) e executam as ações de um plano selecionado. As percepções e ações envolvem buscas, modificações na base de dados do Sig2m e a comunicação com outros AGTs da sociedade.

4.5.1.2.1 A classe TutorManagerAgent

A classe *TutorManagerAgent* é a classe mãe do AGT e constitui a camada de controle, apresentada na Figura 30. A partir dela, são feitas as percepções e ações no ambiente. As ações compreendem o envio de mensagens ao usuário, mensagens a outros AGTs da sociedade, cálculos e análise sobre o estado do estoque. As percepções, por sua vez,

compreendem o monitoramento do estado do estoque, o recebimento de mensagens vindas de outros AGTs, o monitoramento sobre as ações do usuário e a modificação do estoque.

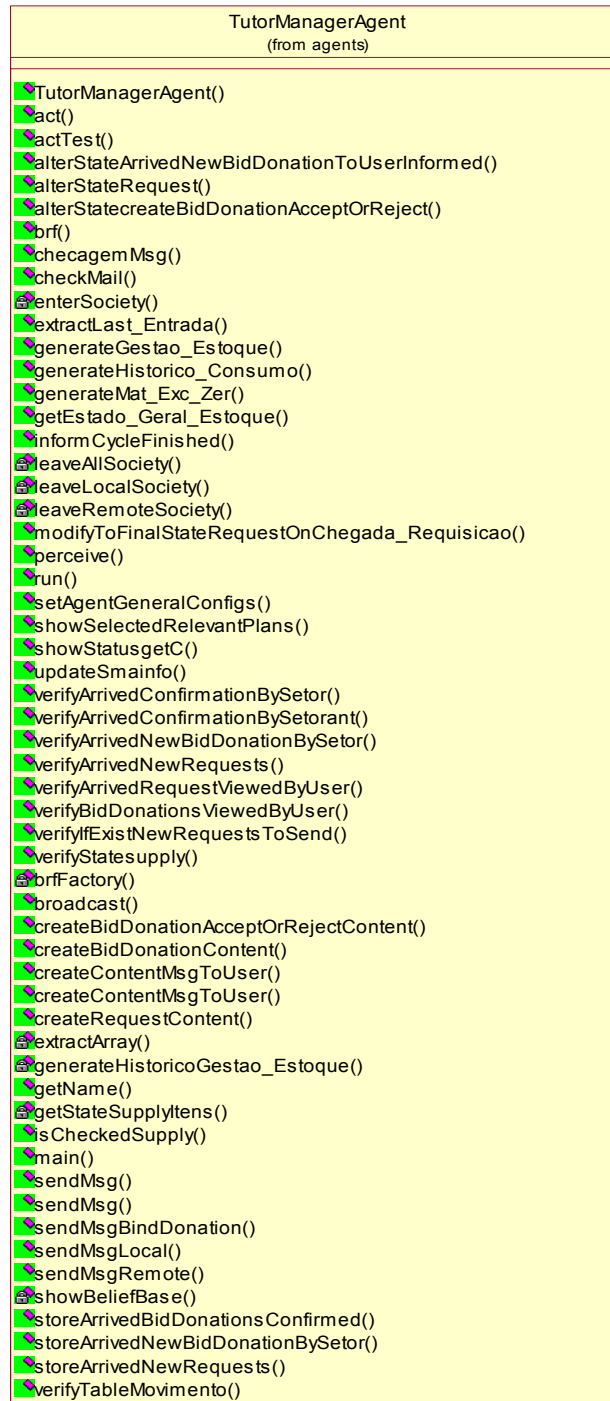


Figura 30 - Classe Java Agente Gestor Tutor – TutorManagerAgent.

Ciclo de execução da classe *TutorManagerAgent* e ligação com a camada cognitiva

A classe *TutorManagerAgent* implementa a interface *AgentArchitecture* que contém os métodos *perceive()*, *brf()* e *act()*, responsáveis, respectivamente, pela percepção, atualização de crenças e atuação no ambiente, conforme Figura 31. A classe *TutorManagerAgent* se associa a uma classe do interpretador Jason, chamada *TransitionSystem*. Esta classe é o elemento de ligação entre o código *AgentSpeak(L)* e a classe *TutorManagerAgent*, que tem por função adicionar e remover crenças no código *AgentSpeak(L)*, interpretar o código e gerar uma ação para que a classe *TutorManagerAgent* execute. Outra funcionalidade da classe *TransitionSystem* é coordenar todo ciclo de funcionamento do AGT: percepção, atualização de crenças e atuação.

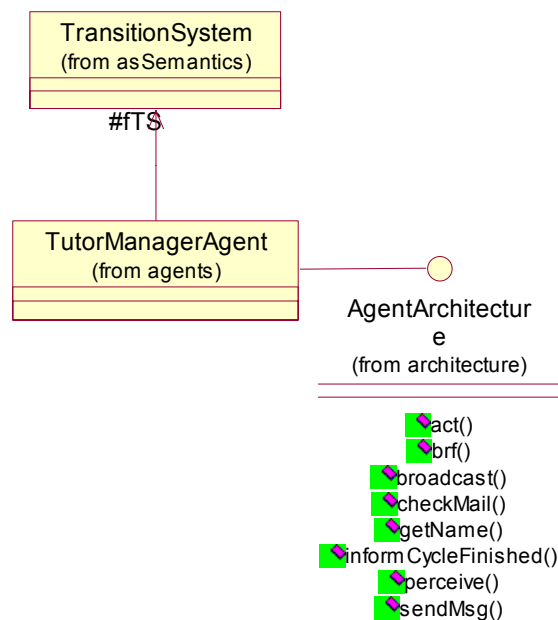


Figura 31 - Diagrama de Classe apresenta a ligação entre a classe *TutorManagerAgent* e o interpretador Jason.

Funcionamento da classe *TutorManagerAgent*

O processo de funcionamento da classe *TutorManagerAgent* é disparado quando é executado o método *run()*. Este método contém a primeira ação do AGT, que é entrar nas

sociedades local e remota. Em seguida, é iniciado o ciclo de execução onde são invocados sequencialmente os métodos *perceive()*, *brf()*, *act()*, *checkMail()* e *sendMessage()*, apresentados a seguir.

4.5.1.2.2 Detalhamento das funções do AGT

A classe *TutorManagerAgent*, contém duas classes de métodos, conhecidos como métodos de percepção e ação.

As percepções do AGT – Métodos de Percepção

As percepções são realizadas pelo método *perceive()*, que invoca os *métodos de percepção*. Estes, por sua vez, fazem verificações sobre o estoque e, através dos resultados obtidos nestas é gerado um conjunto de crenças sobre o ambiente. A Figura 32 apresenta o caso de uso das principais percepções do AGT que contém os processos de verificação sobre:

- Existência de mensagens enviadas por outros AGTs, como por exemplo, a existência de novas requisições de material vindas de outras UBSs;
- Verificar se foi realizado o monitoramento do estoque;
- Verificar se expirou o tempo de monitorar o estoque;
- Verificar se foi realizado alguma ação por parte do usuário com relação à chegada de mensagens vindas de outras UBSs;
- Verificar o estado geral do estoque.

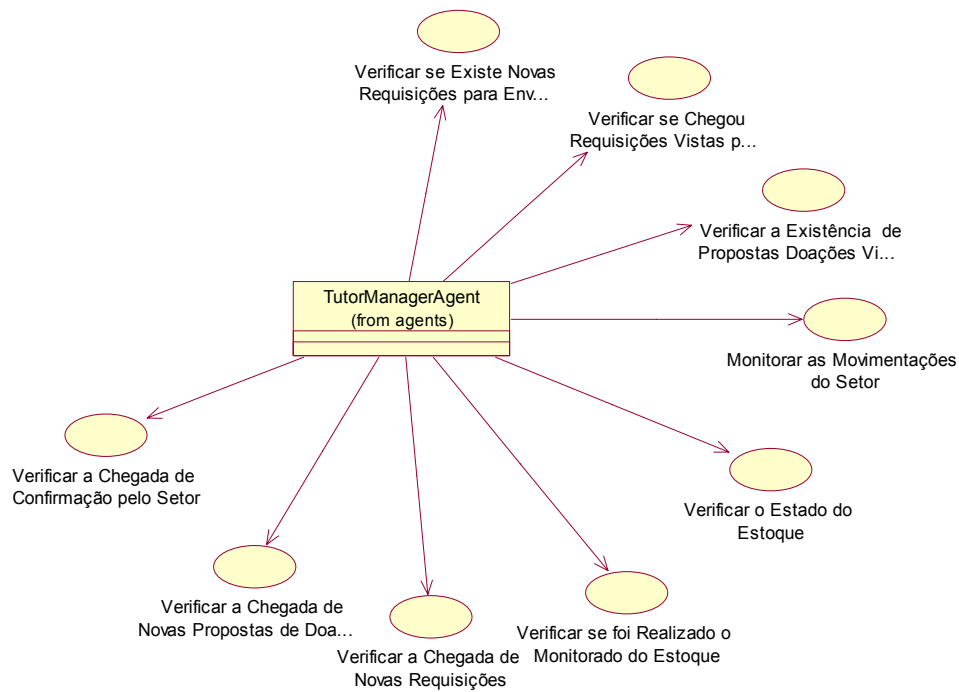


Figura 32 - Caso de uso que descreve as principais percepções realizadas pelo AGT.

Um processo (percepção) é representado por uma classe gerenciadora. As classes gerenciadoras possuem o seu grupo de classes, o qual processa a informação, obtendo o resultado de uma percepção. A Figura 33 representa o diagrama de classes das principais classes gerenciadoras do AGT. Cada processo contém um subdiagrama de caso de uso e seu respectivo diagrama de classe, encontrado no anexo C deste trabalho.

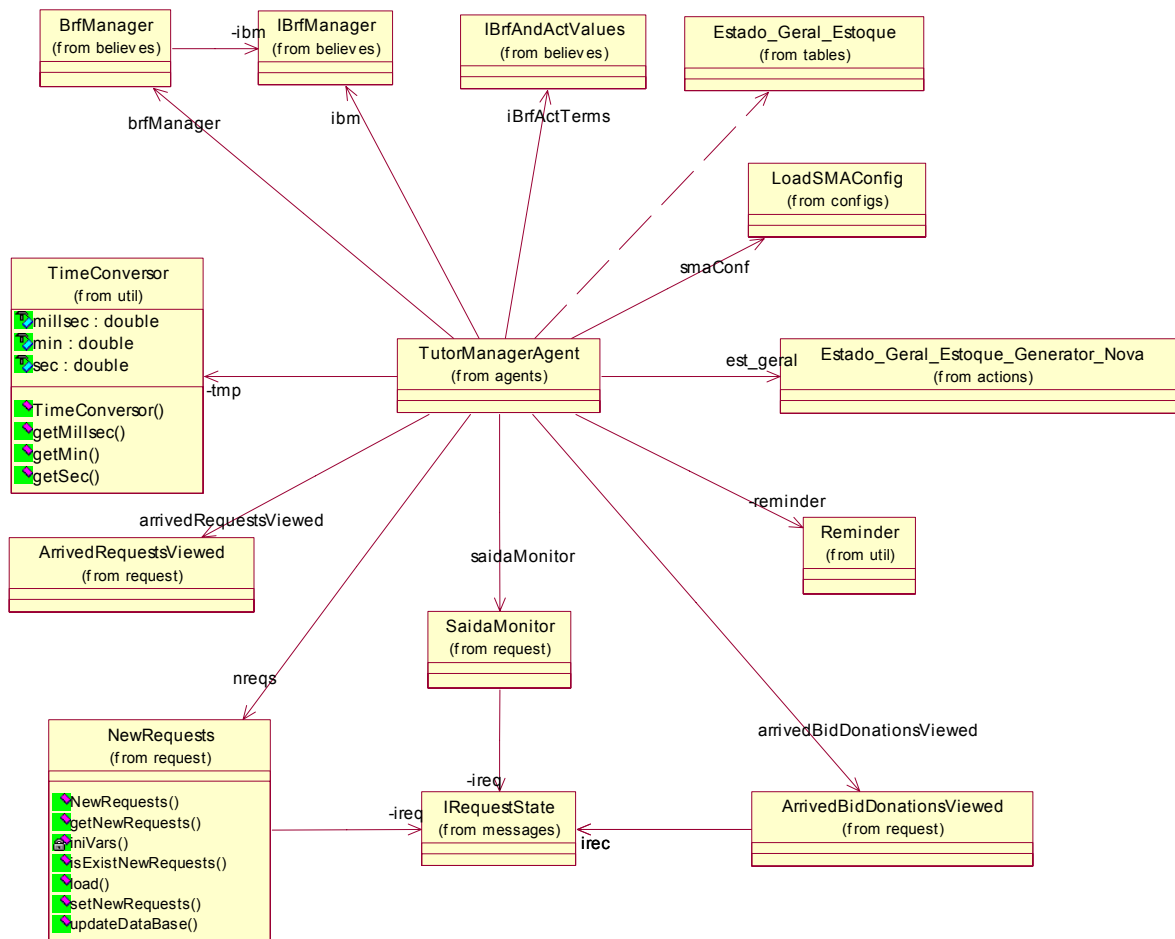


Figura 33 - Diagrama de classes das percepções do AGT.

Métodos de percepção

Os métodos de percepção são responsáveis por monitorar o banco de dados do AGT (estado interno) e do Sig2m, que é o seu ambiente de atuação. No banco de dados do Sig2m, os métodos de percepção monitoram as tabelas que armazenam as movimentações de entrada e saída de materiais. Cada método de percepção gera a adição de uma crença (em formato *AgentSpeak(L)*), quando seu resultado de verificação retorna uma informação verdadeira ou a remoção de uma crença, quando é retornado um valor falso. A Tabela 1 apresenta as assinaturas dos métodos de percepção, sua descrição e sua respectiva crença gerada do AGT.

Métodos de Percepção (Java)	Crença gerada (<i>AgentSpeak(L)</i>)	Descrição
verifyIfExistNewRequestsToSend();	exist(new_request)	Verifica se existe novas requisições do usuário a serem enviadas para os AGTs da sociedade.
verifyArrivedRequestViewedByUser();	exist(bid_donation)	Verifica se o usuário gerou uma proposta de doação para um setor requisitante, a partir de uma requisição do setor.
verifyBidDonationsViewedByUser() ;	exist(bid_donation_accept_reject)	Verifica a existência de confirmação de propostas a doação de materiais pelo usuário.
verifyTableMovimento();	exist(sector_supplied)	Verifica se o setor requisitante já está suprido por uma doação. Modifica o estado final da requisição.
verifyStatesupply();	time_verify_state_supply(true)	Verifica se é tempo de verificar o Estado Geral do Estoque.
getStateSupplyItens(iBrfActTerms.material_excess);	state_supply_material_excess (material_excess)	Verifica se o estoque está em estado crítico com relação aos materiais excedentes.
getStateSupplyItens(iBrfActTerms.material_loosers);	state_supply_material_loosers (material_loosers)	Verifica se o estoque está em estado crítico com relação aos materiais em véspera de vencimento.
getStateSupplyItens(iBrfActTerms.material_zeroed);	state_supply_material_zeroed (material_zeroed)	Verifica se o estoque está em estado crítico com relação aos materiais que irão zerar antes do prazo de suprimento.
isCheckedSupply();	schedule(true):	Verifica se é tempo de realizar os cálculos de gestão do estoque.
verifyArrivedNewRequests() ;	arrived(new_request)	Verifica sobre a chegada de novas requisições, vindas dos setores requisitantes (UBSs).
verifyArrivedNewBidDonationBySetor();	arrived(bid_donation)	Verifica se chegaram novas propostas de doações enviadas pelas UBSs.
verifyArrivedConfirmationBySetor();	exist(confirmation_by_sector)	Verifica se existe confirmação de aceitação ou rejeição a doação de matérias pelo setor requisitante (UBS).
verifyStateEGE();	state_EGE(type_verification, State)	Verifica sobre o Estado Geral do Estoque.

Tabela 1 – Descrição dos Métodos de Percepção e suas respectivas crenças.

A atualização de Crenças

Depois de realizado o processo de percepção é feita a atualização das crenças em que crenças geradas pelos métodos de percepção são incorporadas no código *AgentSpeak(L)*, através do método *brf()*. O método *brf()* invoca métodos da classe *TransitionSystem*, conhecida como *Função de Revisão de Crenças (FRC)*, que adiciona e remove as crenças no código *AgentSpeak(L)*. A partir da base de crenças atualizada, o AGT seleciona uma estratégia (plano) ou um conjunto de planos de sua biblioteca, para atingir os seus objetivos. Um plano é constituído de um conjunto de ações. Para que um plano funcione, é necessário que o agente execute todas as suas ações com êxito. Depois de selecionado um plano, o agente executa as suas ações. A seleção de planos é realizada de forma seqüencial. Para isto, foi implementado, na camada de controle do AGT, o *subsistema administrador de percepções e planos*, responsável por prover o controle de execução das percepções do ambiente e, conseqüentemente, de um plano selecionado até o fim (seja em seu sucesso ou fracasso). Este subsistema faz o tratamento para que não ocorra a interrupção do plano corrente mediante a adição de um outro plano.

As Ações do AGT – Métodos de Ações

As ações do AGT são processos que implicam na modificação do estado do ambiente (banco de dados do Sig2m) ou no estado do próprio agente. Por exemplo, a ação de comunicar com outros AGTs gera a modificação de seu estado interno. As ações são executadas através da execução de um plano selecionado. As principais ações do AGT estão representadas na Figura 34, em que cada processo de caso de uso representa uma ação, que contém um subcaso de uso encontrado no anexo C deste trabalho. As principais ações identificadas do AGT são:

- Entrar na Sociedade Remota: Entra na sociedade Remota;
- Entrar na Sociedade Local: Entra na Sociedade Local;
- Anotar Lista Entrada de Itens: Recupera a última lista de suprimento de materiais, advinda do Almoxarifado;

- Calcular Frequência de Movimento no Estoque: Calcula a frequência de movimentações da lista de itens;
- Calcular Gestão de Estoque: Executa o processo dos cálculos de Gestão do Estoque;
- Calcular a Lista de Materiais que irão Exceder e Zerar: Efetua os cálculos da lista de materiais que irão exceder ou zerar;
- Calcular Estado Geral Estoque: Realiza o cálculo do Estado Geral do Estoque;
- Orientar Usuário sobre Estado Geral Estoque: Depois de realizado o cálculo do Estado Geral do Estoque, o AGT orienta sobre a sua situação ao AGT;
- Comunicar com outros *TutorManagerAgents* Remotos: Comunica com outros AGTs da Sociedade Remota. A comunicação envolve envio e recebimento de requisições e doações de material;
- Comunicar Usuário sobre Mensagens dos *TutorManagerAgents*: Informar o usuário sobre a chegada de mensagens remotas vindas de outros AGTs.

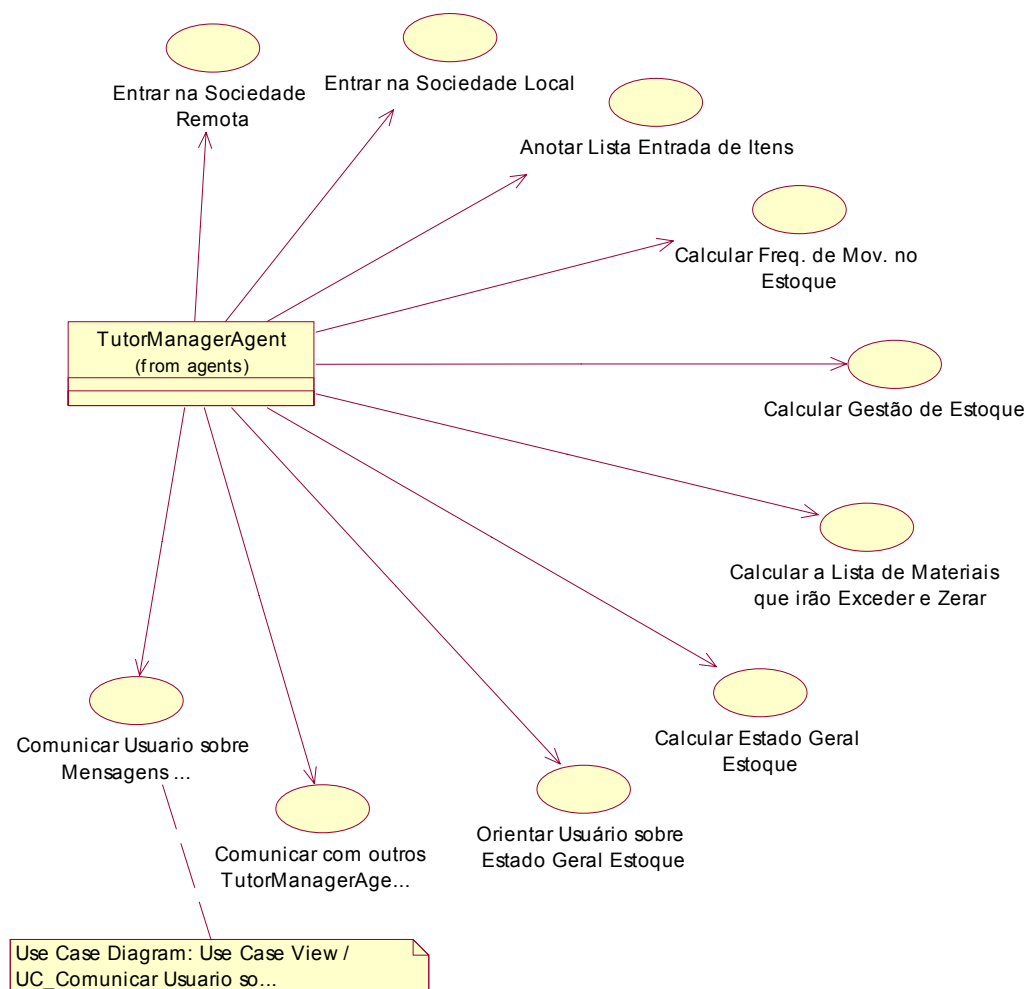


Figura 34 - Caso de Uso das ações do AGT.

A codificação das ações do caso de uso é representada pela Figura 35. As ações possuem a mesma estrutura que as percepções, em que uma classe gerente possui um conjunto de classes para a execução de uma ação e representa um processo do caso de uso.

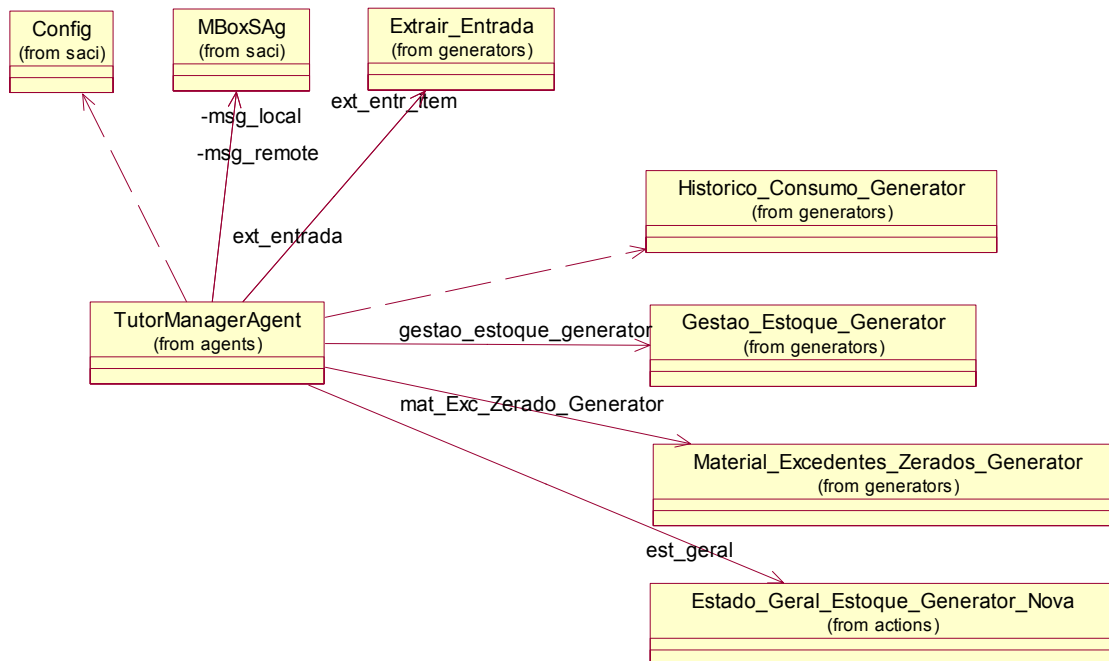


Figura 35 - Diagrama de classes das ações do AGT.

Métodos de ação

Os *métodos de ação* são as ações efetivas que são realizadas pelo AGT. Eles são invocados a partir do método *act()*, em que se recupera uma ação em *AgentSpeak(L)*. Esta ação é interpretada pelo Jason, através da classe *TransitionSystem* e é executada através de um método Java. As ações são apresentadas pela Tabela 2.

Ação gerada (<i>AgentSpeak(L)</i>)	Métodos de Ação (Java)	Descrição
<code>generate_table(extractdata)</code>	<code>extractLast_Entrada()</code>	Extrai a última lista de entrada de itens.
<code>generate_table(historico_consumo)</code>	<code>generateHistorico_Consumo();</code>	Gera Histórico de consumo dos itens.
<code>generate_table(gestao_estoque)</code>	<code>generateGestao_Estoque();</code>	Realiza os cálculos de Gestão do Estoque.
<code>generateMat_Exc_Zer()</code>	<code>generate_table(material_excedente_zerado);</code>	Gera Lista de Materiais que estão excedentes e / ou que irão zerar antes do tempo de suprimento.
<code>update_date_sma(today)</code>	<code>updateSmainfo();</code>	Atualiza a data de verificação sobre o estado do estoque.

calculate(ege)	getEstado_Geral_Estoque();	Realiza o cálculo do Estado Geral do Estoque.
inform_user(material_excess,reasonable_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação tolerável com relação aos materiais excedentes.
inform_user(material_excess,critical_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação crítica com relação aos materiais excedentes.
inform_user(material_loosers,reasonable_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação tolerável com relação aos materiais em véspera de vencimento.
inform_user(material_loosers,critical_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação crítica com relação aos materiais em véspera de vencimento.
inform_user(material_zeroed,reasonable_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação tolerável com relação aos materiais que irão zerar antes do tempo de suprimento.
inform_user(material_zeroed,critical_situation)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário que o estoque está em uma situação crítica com relação aos materiais que irão zerar antes do tempo de suprimento.
inform_user(new_request)	createContentMsgToUser(about); sendMsgLocal(content);	Informa o usuário sobre a chegada de uma nova requisição.
create_content(bid_donation)	createBidDonationContent();	Monta o conteúdo para o envio de uma proposta de doação. O conteúdo da mensagem é um objeto que contém a lista de materiais e a quantidade a supostamente doada de cada material.
send_msg(bid_donation)	sendMsgBindDonation(contents_to_send);	Envia a proposta de doação ao setor requisitante.
inform_user(bid_donation)	createContentMsgToUser(about); sendMsgLocal(content);	Ação para o tratamento de chegadas de proposta de doação. Inicialmente é montada a mensagem e em seguida enviada (envio de uma mensagem local ao Sig2m), informando o usuário sobre a chegada da proposta de doações de materiais.
create_content(bid_donation_accept_reject)	createBidDonationAcceptOrRejectContent();	Gera a mensagem de confirmação de aceitação ou rejeição de doação de materiais.
Send_msg(bid_donation_accept_reject)	sendMsgBindDonation(contents_to_send);	O agente recolhe as doações e as envia a confirmação para cada setor doador.
inform_user(confirmation_by_sector)	createContentMsgToUser(abo	Cria uma mensagem para

	ut) ; sendMsgLocal (content) ;	informar ao usuário sobre a chegada da confirmação de uma proposta ao requisitante. Envia a mensagem ao usuário.
modify_state_request_cicle_to_final(Chegada_Requisicao)	modifyToFinalStateRequestOnChegada_Requisicao() ;	Modifica o estado de uma mensagem de requisição para o estado final de atendida ou não.

Tabela 2 - Descrição das ações geradas em *AgentSpeak(L)* e seus respectivos métodos em Java.

4.5.2 Modelagem matemática do saldo ideal para um estoque

4.5.2.1 Estoque Ideal (EI)

O *Estoque Ideal* (EI) é o nível de estoque que tem por função atender ao propósito de gestão do estoque: atingir o equilíbrio entre o consumo e a demanda de materiais na rede municipal de saúde. O Estoque Ideal serve como comparativo de análise com o estoque real. A partir deste comparativo, é obtida a lista de materiais que estão em estado irregular, ou seja, materiais que estão excedentes e / ou irão atingir a margem de risco de esgotamento antes do tempo de reposição (TR), conhecida como *Lista de Materiais Excedentes e Zerados*.

A situação do estoque considerada ideal constitui a margem de valores compreendida entre o intervalo de excedentes e zerados em um determinado período, conforme ilustrado na Figura 36, cujo estoque ideal é a faixa de valores entre o estoque excedente e o zerado.

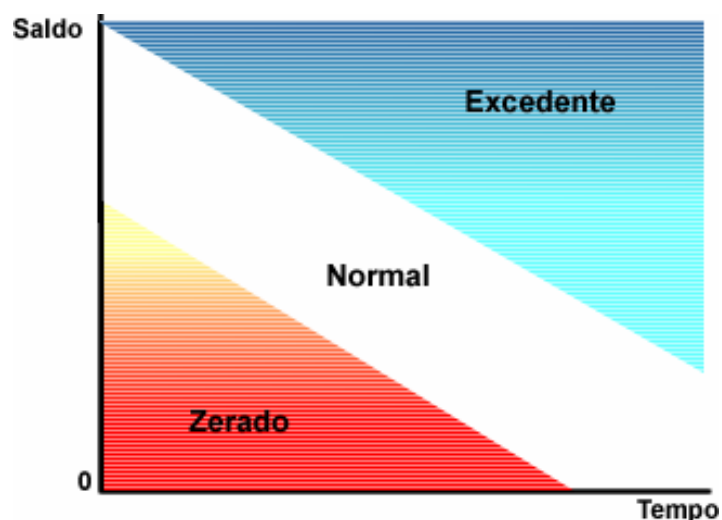


Figura 36– Situação ideal do estoque.

A modelagem matemática do EI de uma UBS foi formulada com base nas fórmulas de gestão de estoque do Estoque Máximo (EMx). A equação de EMx é dada pela fórmula (1) abaixo (ver detalhes no Anexo B):

$$EMx = ES + LC \quad (1),$$

onde ES é o *estoque de segurança* e LC é o *lote de compra*.

Da equação para EI é extraída a quantidade de materiais que estão em pendência (materiais que estão fora do estoque, como materiais encomendados e em inspeção) mostrada pela fórmula:

$$MateriaisPendencia = Encomendas + Inspecao \quad (2)$$

Assim, a fórmula do Estoque Ideal (EI) é dada por:

$$EI = ES + LC - MateriaisPendencia \quad (3),$$

onde $MateriaisPendencia$ é igual ao saldo de materiais em pendência.

4.5.2.2 Cota Ideal (CI)

A *Cota Ideal* (CI) é o valor calculado para cada material a ser requisitado ao *Almoxarifado Central* para que seja suprido no ciclo seguinte. Este processo é realizado por todas as UBSs. Próximo ao fim de cada ciclo, o AGT realiza o cálculo da *Cota Ideal* (CI) e a envia para o *Almoxarifado Central*. O cálculo da *Cota Ideal* é a diferença entre o *Estoque Ideal* e *Estoque Virtual* (EV) (detalhes sobre as fórmulas serão encontrados nos anexo B deste trabalho), ou seja,

$$CI = EI - EV \quad (4)$$

4.5.2.3 A Camada Cognitiva

A camada cognitiva é o núcleo do AGT codificado na linguagem *AgentSpeak(L)*. A partir dela, partem as ordens de execução das ações de um plano do AGT, mediante as percepções do estado do ambiente e do próprio agente. Esta camada contém a codificação dos estados mentais, que são o conjunto de crenças, desejos e intenções do AGT. O núcleo do AGT está dividido em três grupos de código: *Gestão de Estoque*, *Estado Geral do Estoque* e *Comunicação Sociedade Remota*, conforme ilustrado pela Figura 37.



Figura 37 – Constituição do núcleo do AGT.

A *camada de Gestão do Estoque* é o conjunto de planos que realizam os cálculos de gestão de estoque. A *camada de Estado Geral do Estoque* é um conjunto de planos que constituem as estratégias pedagógicas destinadas a orientar o usuário sobre as anomalias do estoque. A *camada de Comunicação Sociedade Remota* constitui o conjunto de planos para gerar o processo de negociação de requisição e doação de medicamentos entre os AGTs da sociedade remota. A seguir, será apresentado com detalhes sobre cada camada do código *AgentSpeak(L)*.

4.5.2.3.1 Codificação *AgentSpeak(L)* - Gestão de Estoque

A camada de Gestão do Estoque contém um conjunto de planos que realizam os cálculos de gestão de estoque a partir da lista de suprimento de materiais. Com os cálculos de gestão de estoque é possível prever quando e quanto adquirir de material para o mês seguinte, e estimar um estoque ideal para cada material. A partir do estoque ideal é possível mensurar a quantidade de itens que estão em estado excedente e / ou que irão zerar antes do tempo de suprimento, possibilitando o seu tratamento.

O conjunto de planos adotados para esta camada estão na Figura 38. Os planos 1, 2 e 3 realizam a verificação se já foi expirado o tempo de realizar os cálculos de gestão do estoque. Em caso afirmativo são disparados os planos de cálculos de gestão de estoque (planos 5, 6, 7, 8 e 9). Os cálculos de gestão de estoque foram programados para serem feitos diariamente.

```

//***** Conjunto de planos para calcular as tabelas de gestão de estoque. ****
+schedule(Expired): (Expired == true)
<-
+expired(date);
+check(extractdata).

+schedule(Expired): not (Expired == true)
<- true.

+expired(date): true
<- true.

+schedule(Expired): not (Expired == true) // remove a crença de data expirada de verificar as
tabelas de dados
<- -expired(date).

```

```

+check(Table): (Table == extractdata) & expired(date) (4)
<- !generate_table(Table).

+!generate_table(Table): (Table == extractdata) (5)
<- generate_table(Table); //gera a tabela extrair dados
+has_calculated(exctract_data); // crença que já foi calculado
!generate_table(historico_consumo).

+!generate_table(Table): (Table == historico_consumo) & has_calculated(exctract_data)
<- generate_table(Table); (6)
+has_calculated(historico_consumo);
!generate_table(gestao_estoque).

+!generate_table(Table): (Table == gestao_estoque) & has_calculated(historico_consumo)
<- generate_table(Table); (7)
+has_calculated(gestao_estoque);
!generate_table(material_excedente_zerado).

+!generate_table(Table): (Table == material_excedente_zerado) &
has_calculated(gestao_estoque) (8)
<- generate_table(Table);
+has_calculated(material_excedente_zerado);
!update_sma_info(today).

+!update_sma_info(Date): has_calculated(exctract_data) & has_calculated(historico_consumo) &
has_calculated(gestao_estoque) & has_calculated(material_excedente_zerado) // se todas as
tabelas foram geradas então atualiza-se a data de verificação das tabelas. (9)
<- update_date_sma(Date).

```

Figura 38 - Estrutura de planos a camada de gestão do estoque.

4.5.2.3.2 Codificação *AgentSpeak(L)* - Estado Geral do Estoque (EGE)

A Figura 39 contém a representação da camada *Estado Geral do Estoque*. Este conjunto de estratégias faz a análise e o tratamento do estoque, dependendo do seu estado. A análise do estoque é feita sobre três índices: índice de materiais excedentes no estoque; índice de materiais que irão zerar no estoque antes do tempo de suprimento; e índice de materiais que possuem lotes em véspera de vencimento.

```

//***** Conjunto de planos sobre informar o usuário sobre o estado do estoque *****

+time_verify_state_supply(Expired): not (Expired == true) (10)
<- // apenas para concluir o plano em que o tempo de verificar o estoque não foi expirado.

true.

+time_verify_state_supply(Expired): (Expired == true) (10)
<- // em seguida será acrescentada uma crença (state_supply(Value)) contendo o cálculo do ege.
   calculate(ege).

+state_EGE(Type_verification, State): (Type_verification == EGE) (11)
<- inform_user(Type_verification, State).

+state_supply_material_excess(Type_verification, State): (Type_verification == material_excess)
<- inform_user(Type_verification, State). (12)

+state_supply_material_loosers(Type_verification, State): (Type_verification == material_loosers)
<- inform_user(Type_verification, State). (13)

+state_supply_material_zeroed(Type_verification, State): (Type_verification == material_zeroed)
<- inform_user(Type_verification, State). (14)

+more_material_list (Type_verification, Percentual): (Type_verification == arrived_point_order)
<- inform_user(Type_verification, Percentual). (15)

+more_material_list (Type_verification, Percentual): (Type_verification == arrived_security_supply)
<- inform_user(Type_verification, Percentual). (16)

```

Figura 39 - Estado Geral do Estoque.

Os dois primeiros planos apresentados constituem um conjunto de estratégias para verificar se o tempo de calcular o estado do estoque já foi expirado. Se o termo *Expired* está com valor *true*, é adotada a ação de calcular o EGE do estoque (*calculate(ege)*). Realizado o cálculo do EGE, é adicionada a crença (*state_supply_material_excess*) na biblioteca de crenças do agente sobre o estado dos materiais que estão excedentes no estoque. O estado dos materiais excedentes representa o percentual em relação à quantidade total de itens da lista de suprimento. O termo *State* informa a gravidade da situação do parâmetro em questão e pode assumir, basicamente, três valores: Estável, Razoável, e Crítico. Dependendo da

gravidade do termo *State*, o AGT adota a estratégia pedagógica mais adequada à situação. As estratégias pedagógicas constituem ações de envio de mensagens ao usuário. São indicativos que o informam sobre a anomalia do estoque e o instrui como proceder para o seu tratamento. As mensagens compreendem em: mensagens informativas, para as situações do estoque em que seu estado é estável; mensagens de advertência que adverte o usuário sobre algum parâmetro que está em estado razoável; e mensagens de alerta, que apontam um estado crítico do estoque.

O plano 13 (*state_supply_material_loosers*) verifica o estado dos itens que estão em véspera de vencimento. Em caso afirmativo, é adotada a mesma estratégia do plano anterior, que é informar o usuário sobre do estoque.

No plano 14 é realizada a verificação do estado dos itens que estão zerados, e é adicionada a crença *state_supply_material_zeroed*. Esta estratégia segue a mesma estrutura que as anteriores de informar o usuário sobre o seu estado.

As duas últimas estratégias (planos 15 e 16) referem-se ao saldo em estoque da lista de materiais. Se a maior parte da lista atingiu o *ponto de pedido* (PP) ou o *estoque de segurança* (ES), o usuário é informado para tomar as providências necessárias de fazer a requisição dos itens em falta, quando são obtidas as crenças de *more_material_list(arrived_point_order, Percentual)* e *more_material_list(arrived_security_supply, Percentual)*. Maiores detalhes sobre o PP e ES e os cálculos de gestão de estoque estão no anexo B.

4.5.2.3.3 Codificação *AgentSpeak(L)* - Comunicação com a Sociedade Remota

A camada de comunicação representa o conjunto de planos do protocolo de negociação de requisição e doação de medicamentos da sociedade remota. Esta camada contém a codificação *AgentSpeak(L)* desde as estratégias de envio de requisição até a efetiva saída do determinado material requisitado por parte do doador, vide a Figura 40.

Os quatro primeiros conjuntos de planos são utilizados por um AGT requisitante de medicamentos. Tem por função realizar a verificação e envio de requisições de medicamentos para a sociedade remota.

O primeiro plano tem por finalidade verificar se foram geradas novas requisições. Este plano é selecionado mediante a adição da crença *verify(new_request)*. Nele, é executada a ação básica de checar a existência de alguma nova requisição. A checagem certifica se o usuário realizou alguma nova requisição para a sociedade remota. A verificação é feita pela leitura na base de dados do AGT, que contém as tabelas do processo de requisição e doação dos medicamentos. Caso exista uma nova requisição, é adicionada a crença *exist(new_request)* na biblioteca de crenças do AGT, que dispara o plano 18. A estratégia 18 é composta por dois elementos a serem executados: o primeiro é uma ação (*create_content(new_request)*) cujo AGT deve realizar a construção do conteúdo da requisição; e a segunda é a chamada de um outro plano (*!send_remote_message(new_request)*), o qual tem por função realizar o envio da requisição de materiais à sociedade remota para que, de alguma forma, possa ser atendida. O agente facilitador da sociedade remota cuida de encaminhar tal requisição para todos os AGTs da sociedade.

Quando os AGTs, supostos doadores, recebem a requisição de materiais, esta é armazenada em suas bases de dados e, em seguida, é adotado o processo de verificação da chegada desta requisição, pelo plano 21. Caso tenha chegado a requisição, é acrescentada a crença *arrived(new_request)* e é realizada a ação (*inform_user(new_request)*) de informar ao usuário sobre a sua chegada, através do envio de uma mensagem para a sociedade local. Se o usuário tiver realizado uma proposta de doação para o setor requisitante, então a estratégia 22 é executada, mediante a adição da crença *exist(bid_donation)*. Note que o plano 22 é análogo à estratégia de envio de requisição (plano 18) e tem por função enviar uma proposta de doação de material para o setor requisitante.

Quando o AGT requisitante recebe as propostas de doações advindas dos supostos doadores e realiza a ação de informar o usuário sobre a sua chegada, descrita no plano 24, então o usuário que realizou a requisição pode decidir em aceitar ou rejeitar às propostas de doações. Em seguida, o AGT realiza a ação de enviar tal confirmação aos candidatos doadores, conforme o plano 25. Para os setores selecionados a doação, é adotado plano 27, que informa o usuário sobre a confirmação do setor requisitante para que possa realizar o processo de doação de tal medicamento. O plano 28 constitui a finalização do processo de doação, ponto em que foi gerada a doação do material para o setor requisitante.

```

//***** Conjunto de Planos do Sistema de Requisição e Doação de Materiais *****

//***** Conjunto de planos para a verificar e enviar as requisições *****
+verify(Type): (Type == new_request) 17
<- check(Type).

+exist(Type): (Type == new_request) 18
<- create_content(Type);
!send_remote_message(Type).

+exist(Type): (Type == no_new_request) 19
<- true.

//***** Conjunto de planos para a verificar e envio de msgs Remotas *****
+!send_remote_message(Type): (Type == new_request) 20
<- send_msg(Type).

//***** Conjunto de planos para a verificar A Chegada de Requisições *****
+arrived(Type): (Type == new_request) 21
<- inform_user(new_request).

//** Conjunto de planos para envio de Proposta de Doação (Atendimento_Requisicao) ao requisitante
***
+exist(Type): (Type == bid_donation) 22
<- create_content(Type);
!send_remote_message(Type).

//***** Conjunto de planos para a verificar e envio de msgs Remotas *****
+!send_remote_message(Type): (Type == bid_donation) 23
<- send_msg(Type).

//***** Conjunto de planos para a verificar A Chegada de Proposta de Doação *****
+arrived(Type): (Type == bid_donation) 24
<- inform_user(bid_donation).

//** Conjunto de planos para envio de Confirmacao da Proposta de Doação (Atendimento_Requisicao)
ao doador ***
+exist(Type): (Type == bid_donation_accept_reject) 25
<- create_content(Type);
!send_remote_message(Type).

//***** Conjunto de planos para a verificar e envio de msgs Remotas para o doador *****
+!send_remote_message(Type): (Type == bid_donation_accept_reject) 26
<- send_msg(Type).

//**Conjunto de planos de CONFIRMACAO DE ACEITACAO OU REJEICAO a doacao do requisitante.**
+exist(Type):(Type == confirmation_by_sector) 27
<- inform_user(confirmation_by_sector).

//***** Conjunto de planos de Modificação do estado final das requisições pelo agente doador *****

```

```
+exist(sector_supplied): true  
<- modify_state_request_cicle_to_final(Chegada_Requisicao).
```

28

Figura 40 - Estrutura de planos para o processo de requisição e doação para a sociedade remota.

4.5.2.4 A Camada de Comunicação

A camada de comunicação representa “fala e audição” do AGT. Esta camada é responsável por gerir a comunicação com outros agentes da sociedade. Possui uma biblioteca de mensagens definida para cada situação encontrada no cotidiano das sociedades, bem como mensagens para instruir o usuário sobre o estado do estoque.

A comunicação entre os agentes é assíncrona e é feita através de um protocolo de comunicação. O protocolo de comunicação adotado é o *Knowledge Query and Manipulation Language* (KQML), que dá suporte a alto nível de comunicação entre agentes, para realizar a troca de conhecimento e informações [WEI1999] [ALV1997].

A camada de comunicação foi construída com a utilização dos componentes da ferramenta *Simple Agent Communication Infrastructure* (SACI), um *framework Open Source*, implementado em JAVA pelo Laboratório de Técnicas Inteligentes (LTI), do Departamento de Engenharia de Computação e Sistemas Digitais (PSC) da Universidade de São Paulo USP. Tem como enfoque principal habilitar a comunicação entre agentes distribuídos em uma rede, através do protocolo de comunicação KQML conforme Finin [FIN1993].

Hübner [HUB2003a] [HUB2003b] relata que o SACI é uma ferramenta que torna transparente para o programador de agentes os aspectos de comunicação distribuída, tendo como principais objetivos ser simples de utilizar e ter um bom desempenho.

O SACI possui as seguintes características:

- Os agentes utilizam KQML para se comunicar. Há funções para compor, enviar e receber mensagens KQML.
- Os agentes são identificados por um nome. As mensagens são transportadas utilizando-se somente o nome do receptor, cuja localização na rede é transparente.
- Um agente pode conhecer os outros por meio de um serviço de páginas amarelas. Agentes podem registrar seus serviços no facilitador e perguntá-lo sobre que serviços são oferecidos por quais agentes.

- Os agentes podem ser implementados como *applets*⁸ e ter sua interface em uma *home-page*.
- Os agentes podem ser iniciados remotamente.
- Os agentes podem ser monitorados. Os eventos sociais (entrada na sociedade, saída, recebimento ou envio de mensagens) podem ser visualizados e armazenados para análise futura.

Hübner [HUB2003a], faz uma comparação da ferramenta SACI com outras ferramentas de comunicação entre agentes em que o SACI apresentou bons resultados em termos de desempenho, em relação às demais ferramentas.

O sistema de envio e recebimento de mensagens é feito por componentes SACI, conhecidos como *MBoxSAg*, que servem de interface entre o agente e sua sociedade. O processo de envio e recebimento de mensagens ao longo da rede é feito de maneira transparente ao usuário.

As mensagens locais e remotas às sociedades são enviadas pelos métodos *sendMsgRemote(Object ob)* e *sendMsgLocal(Object message)* do AGT. A Figura 41 apresenta o diagrama de classe de envio de mensagens locais e remotas do AGT. As mensagens KQML são construídas através de uma classe fábrica chamada *MessageFactory*. A *MessageFactory* se relaciona com as classes *KQML_Comm_Layer*, *KQML_Msg_Layer* e *KQML_Content_Layer*, sendo que cada uma é responsável por produzir uma camada da estrutura da mensagem. A classe *KQML_Comm_Layer* gera a camada de comunicação, ela produz a descrição do remetente e destinatário da mensagem. A classe *KQML_Msg_Layer* define a linguagem e ontologia para que o destinatário possa interpretar o conteúdo da mensagem. Já a classe *KQML_Content_Layer* é responsável por gerar o conteúdo da mensagem, a informação a ser enviada.

⁸ *Applets* são aplicativos que têm por finalidade adicionar interatividade a aplicações web. Utilizam a JVM (*Java Virtual Machine*), existente na máquina cliente ou embutida no próprio navegador do cliente.

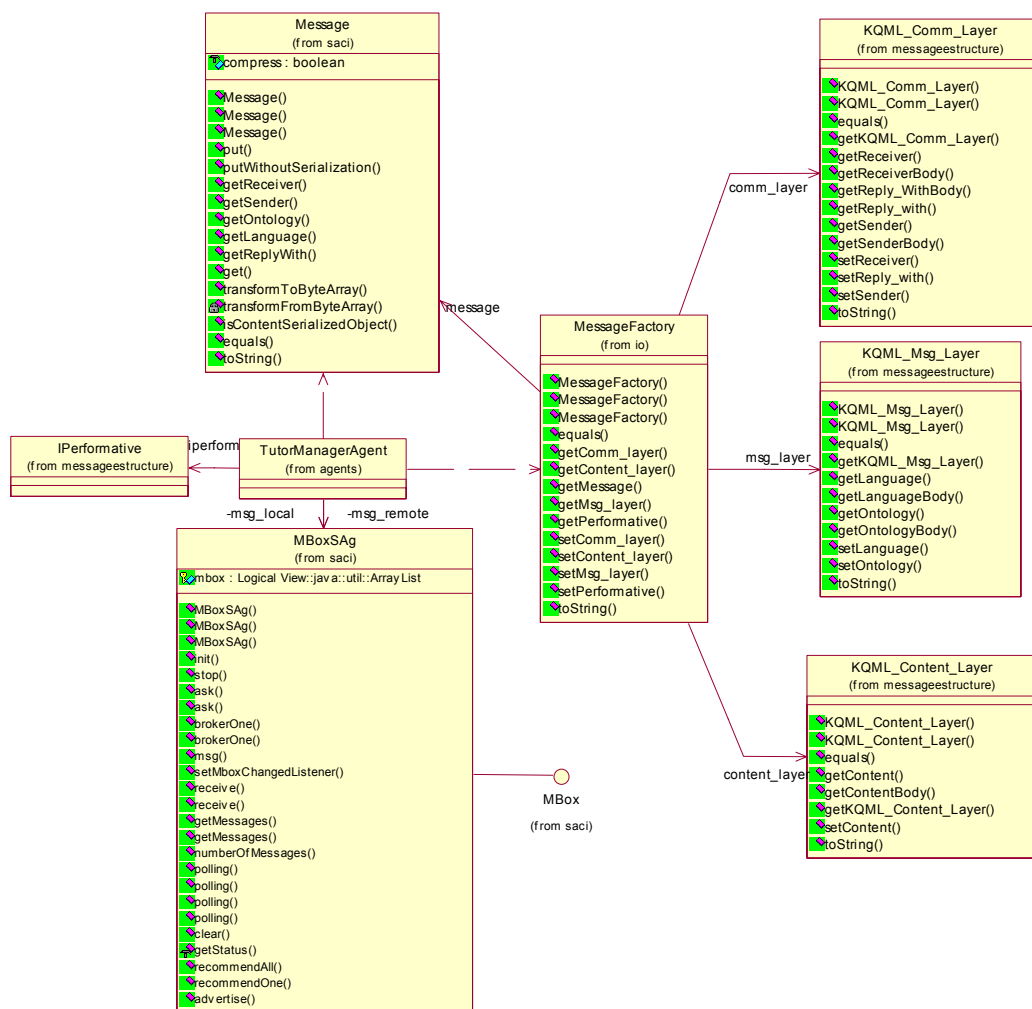


Figura 41 - Diagrama de classe de envio e recebimento de mensagens locais e remotas do AGT.

A cada ciclo de execução do AGT é feita a checagem no *MBoxSag* a respeito da chegada de mensagens advindas de outros agentes, ele é invocado através do método *checkMail()*. Na chegada de uma mensagem, o AGT extrai o seu conteúdo e a armazena no seu banco de dados para em seguida informar o usuário.

4.5.2.4.1 Modelagem do Agente Mensageiro - AM

O *Agente Mensageiro* (AM) é um agente reativo que tem por função gerar o canal de comunicação, uma interface entre o AGT e o usuário através do Sig2m, vide Figura 42. A comunicação entre os agentes AGT e o AM é realizada pelo protocolo de comunicação KQML.

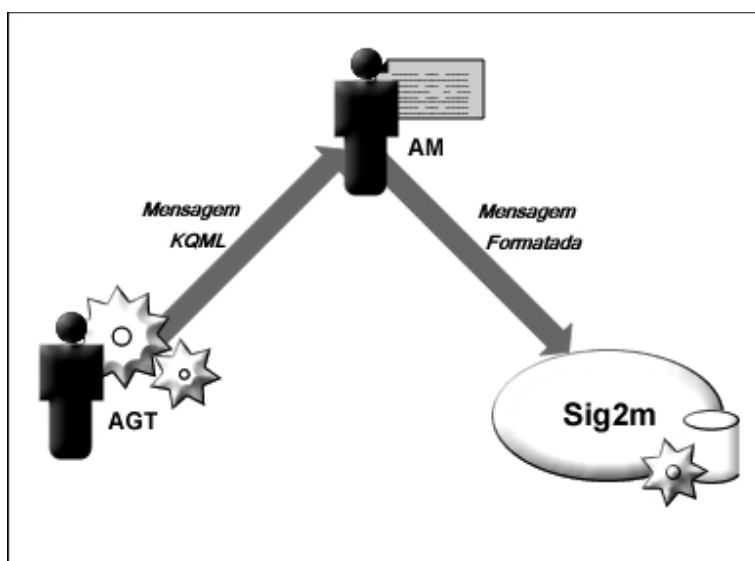


Figura 42 - Arquitetura do Agente Mensageiro.

A Figura 43 apresenta o caso de uso do AM com as suas principais funções, o AM é representado pela entidade *LocalAgent*. A tarefa inicial do AM é entrar na sociedade local para poder receber mensagens do AGT. O processo de entrada na sociedade consiste em o AM informar ao agente facilitador da sociedade sobre a sua existência. Após ter entrado na sociedade local, o AM aguarda por mensagens vindas do AGT, e quando as recebe, ele extrai o seu conteúdo, formata-o de forma inteligível para o usuário e o apresenta através do Sig2m, em forma de uma caixa de diálogo.

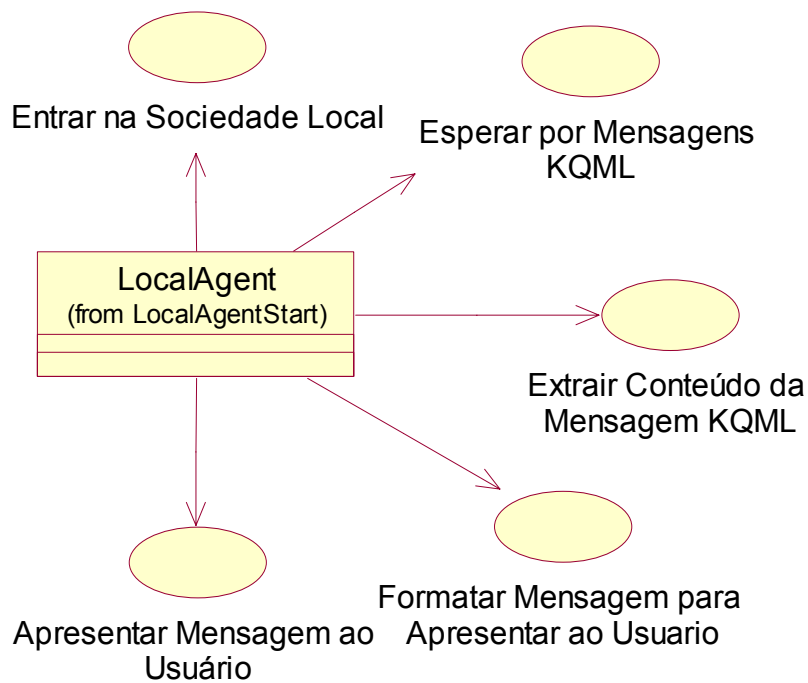


Figura 43 - Diagrama de caso de uso do Agente Mensageiro.

O AM é uma classe estendida da classe *Agent* pertencente a ferramenta SACI, de acordo com a Figura 44. A classe *Agent* é uma classe do tipo *thread* que contém funções para a entrada e saída do agente na sociedade local. O AM se associa com a classe *MBoxSAG*, que provê suporte a recepção e envio de mensagens KQML. As mensagens do AGT são interceptadas pelo método *receptorMsgs()*. O seu conteúdo é extraído pelo método *extractData()* e formatado pelo método *messageFormater()* e é enviado ao Sig2m pelo método *senderMsgs()*.

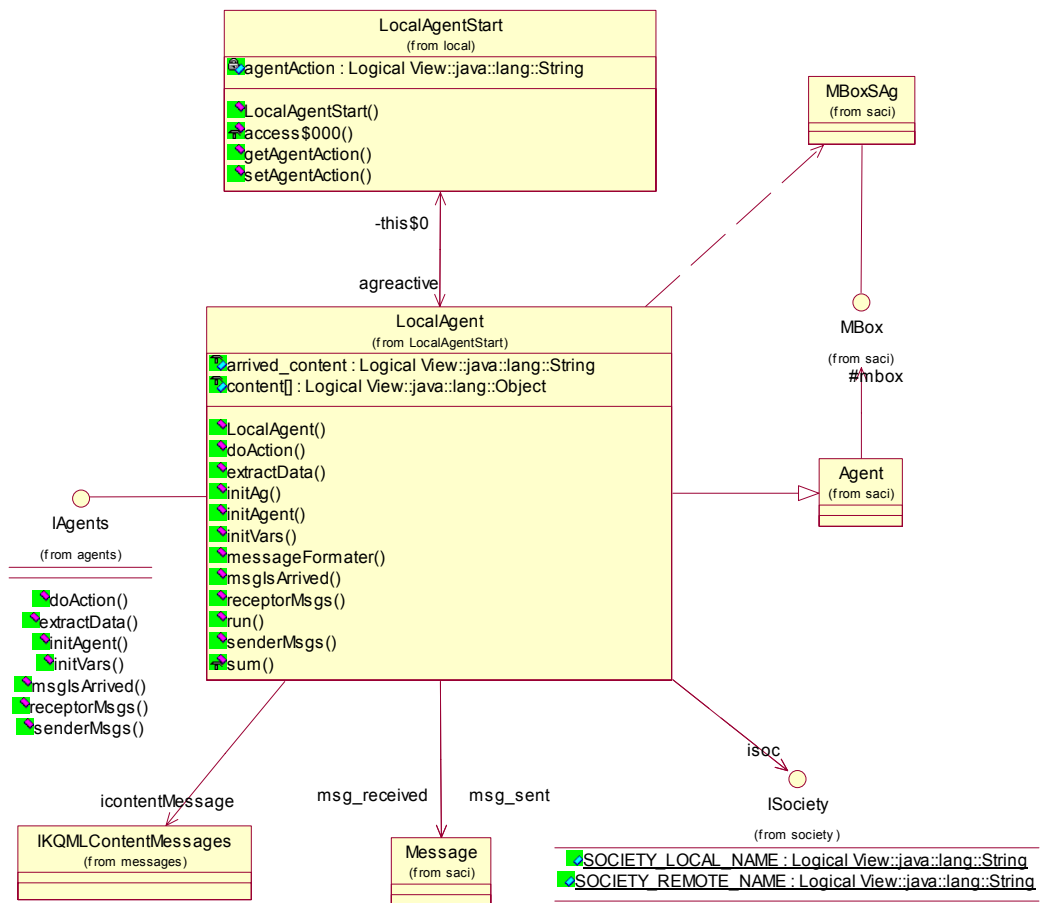


Figura 44 - Diagrama de classes do AM.

O AM se encontra como *thread* na mesma Máquina Virtual Java (JVM) que o Sig2m, apresentado pela Figura 45. Quando o Sig2m é executado, o AM é disparado através da classe *LocalAgentStart* que contém uma referência à classe *LocalAgent*.

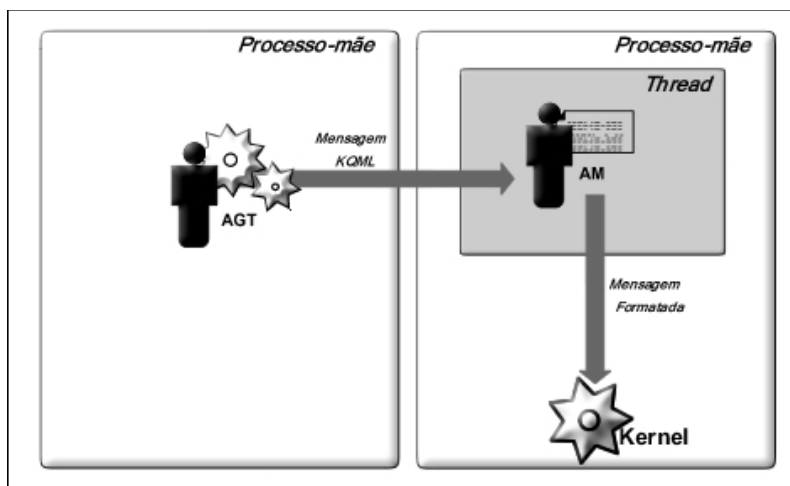


Figura 45 - Disposição dos Processos AGT, AM e Sig2m.

4.6 Componentes da Arquitetura Clássica de um STI sobre Arquitetura Implementada

A arquitetura clássica de um STI está disposta na arquitetura do AGT, conforme ilustrado pela Figura 46. O AGT monitora as ações realizadas pelo usuário no Sig2m, através das tabelas de movimentações de materiais e realiza os cálculos de gestão para obter um estoque ideal para cada material. O AGT armazena as informações dos cálculos de gestão e do estoque ideal em seu banco de dados. A partir destas informações, ele analisa a situação do estado do estoque e orienta o usuário, com a finalidade de atingir um patamar ideal para atender às necessidades das UBS. A seguir, será enfatizado com mais detalhes cada componente da arquitetura e suas funções.

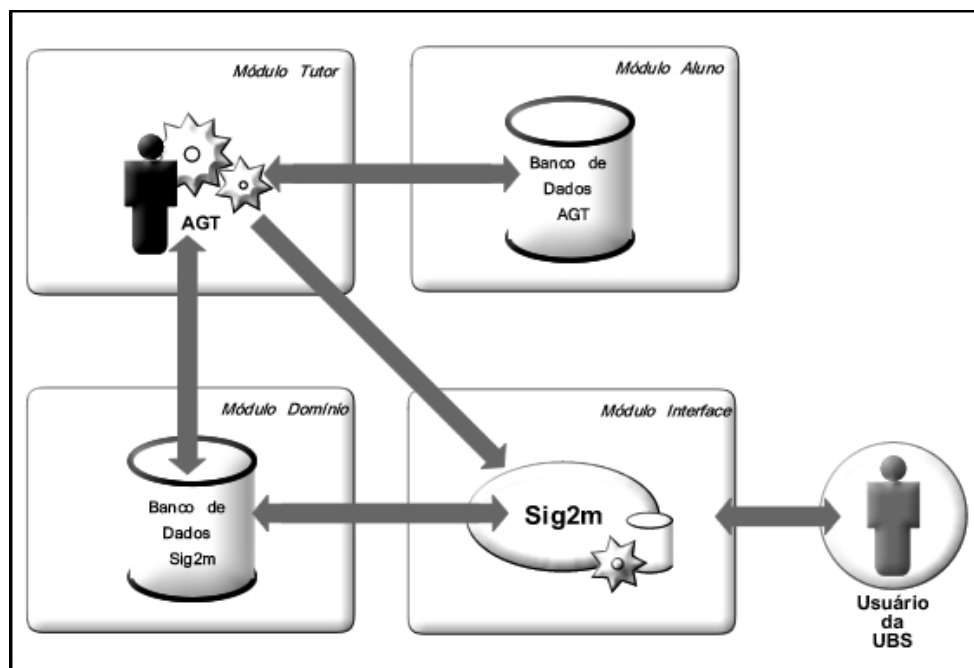


Figura 46 - Disposição dos componentes da arquitetura do STI sobre a arquitetura implementada.

4.6.1 Módulo de Domínio

O *Módulo de Domínio* está disposto como o modelo de aspectos do conhecimento sobre o domínio que o aprendiz pode acessar. Ele armazena o conteúdo a ser trabalhado com o aprendiz. O *Módulo de Domínio*, na arquitetura implementada, representa as informações sobre todo o estoque em que o usuário pode acessar pelo Sig2m. Estas informações possibilitam o usuário a realizar operações no estoque, na tentativa de atingir um patamar ideal para o mesmo.

4.6.2 Módulo do Tutor

O *Módulo do Tutor* é representado pelo AGT, conforme ilustrado pela Figura 46. Ele é o detentor das estratégias de auxílio ao usuário. Seu principal objetivo é instruir o usuário a realizar ações que refletem no EGE a atingir o caso ideal, ou seja, todos os parâmetros do EGE tender a zero, lembrando que o cálculo do EGE é modelado sobre os materiais em

estado de anomalia, que constam na lista de materiais excedentes, zerados e que possuem lotes em véspera de vencimento.

O modo de instrução ao usuário, implementado no AGT, é o *modelo de treinamento*. Neste modelo, o STI observa o desempenho do usuário e apresenta conselhos que o ajudarão a executar uma dada atividade da melhor forma possível. Esta estratégia inclui uma orientação para tarefas novas e interação com o aprendiz para achar as melhores opções para a resolução de uma dada atividade.

Estratégias de tratamento do estoque do AGT

As estratégias de tratamento do estoque feitas pelo AGT constituem em instruir o usuário a:

- Gerar saídas para os materiais que estão excedentes, seja para o almoxarifado gestor ou para as UBS;
- Gerar movimentações de entrada de materiais para os itens que estão zerados;
- Alertar o usuário sobre itens que atingiram o nível de ponto de pedido (PP) e também o estoque de segurança (ES);
- Realizar negociações com outros AGTs da sociedade, a fim de proverem, em conjunto, uma solução cooperativa para a distribuição dos materiais na rede.

4.6.3 Interface

A interface é responsável por gerar a interação entre o tutor e o aprendiz. Na arquitetura disposta, a interface foi implementada em dois sentidos de fluxos de informação:

- Direto: A interação entre o AGT e o usuário (tutor – usuário) é realizada através da sociedade local, ora discutida. Em que o Sig2m é utilizado como meio de comunicação.
- Indireta: Interação realizada entre o usuário e o AGT (usuário - tutor). Este tipo de interação é feito mediante as ações em que o usuário gera, as quais consistem em, operações de movimentação de estoque pelo Sig2m. As movimentações são

captadas pelas percepções do AGT ao observar a base de dados que contém todo o conhecimento da informação (ambiente).

4.6.4 Módulo do Aprendiz

O *Módulo do Aprendiz* representa as características individuais do aprendiz, como as suas informações e ações em relação ao *Módulo de Domínio* (o estoque), que repercutem de forma dinâmica.

O módulo do aprendiz não é um conceito trivial de ser implementado, pois ainda é difícil projetá-lo em conformidade com um perfil que contemple, na íntegra, a representação do aprendiz no computador, como avaliar o seu comportamento, suas habilidades, um retorno em relação ao conteúdo, conforme já discutido no capítulo 2.

No contexto da arquitetura discutida, o Módulo do Aprendiz está representado como as ações do usuário realizadas sobre o estoque. Estas ações são quantificadas através do cálculo do *Estado Geral do Estoque* (EGE), feito pelo AGT. O EGE é um conjunto de parâmetros em que é possível analisar o estoque sobre várias óticas e indicar qual a sua situação geral. Através do cálculo do EGE, é possível obter um perfil do aprendiz através de seu comportamento, como ações em seu estoque perante as mais variadas situações do seu dia-a-dia. Assim, o AGT obtém subsídios para adotar a estratégia pedagógica mais adequada para auxiliar o usuário a resolver as anomalias encontradas no estoque. Em suma, o EGE é a representação do Módulo do Aprendiz, composto por um conjunto de parâmetros que indicam as ações realizadas, para que o estoque tenda ao estado ideal.

4.6.4.1 Construção do Módulo do Aprendiz (Estado Geral do Estoque)

O EGE foi modelado com base em três parâmetros. Cada parâmetro analisa o estoque sobre um aspecto distinto. Os parâmetros selecionados para compor o EGE são:

- Lotes em Véspera de Vencimento (LVV): Trata-se da quantidade de lotes que estão próximos ao prazo do vencimento de um material. É um indicativo do comportamento do usuário em relação ao estado do estoque, caso existam lotes que estão para vencer.

- **Materiais Zerados (MZ):** É o indicativo mais agravante para um estoque. Consiste nos materiais que irão zerar antes do prazo de suprimento. Dependendo do tipo de material, como os de classificação Z, sua ausência pode trazer sérias conseqüências tanto a um setor como também para o usuário final.

- **Materiais Excedentes (ME):** Representa a quantidade de itens excedentes no estoque. Existem algumas situações cujos materiais excedentes podem ser um outro indicativo a chamar a atenção em um estoque. Como o contexto deste estudo de caso se trata de uma rede municipal de saúde, é desinteressante possuir um estoque com medicamentos em excesso, o correto é ter uma quantidade suficiente para atender até o período de suprimento (ciclo).

Cada parâmetro é obtido através da média ponderada da quantidade de itens que correspondem a um parâmetro em relação aos pesos atribuídos a cada variável envolvida. Assim, para a *Média de Materiais com Lotes em Véspera de Vencimento* (MMLVV) tem-se a seguinte fórmula:

$$MMLVV = \frac{(QtdeMatLotesVespVencX * P1 + QtdeMatLotesVespVencY * P2 + QtdeMatLotesVespVencZ * P3)}{(P1 + P2 + P3)} \quad (5)$$

onde QtdeMatLotesVespVencX é a quantidade de materiais do tipo de classificação X que possui lotes em véspera de vencimento, a QtdeMatLotesVespVencY é a quantidade de materiais do tipo de classificação Y que possui lotes em véspera de vencimento, QtdeMatLotesVespVencZ representa a quantidade de materiais do tipo de classificação Z que possui lotes em véspera de vencimento e P1, P2 e P3 são pesos para cada variável respectiva.

Analogamente para o parâmetro de materiais zerados em estoque, tem-se a seguinte fórmula, titulada como *Média de Materiais Zerados* (MMZ):

$$MMZ = \frac{(QtdeMatZeradosX * P1 + QtdeMatZeradosY * P2 + QtdeMatZeradosZ * P3)}{(P1 + P2 + P3)} \quad (6)$$

onde $QtdeMatZeradosX$ representa a quantidade de materiais de classificação X que irão ou estão zerados no estoque, $QtdeMatZeradosY$ é a quantidade de materiais de classificação Y que irão zerar ou estão zerados no estoque, $QtdeMatZeradosZ$ trata-se quantidade de materiais de classificação Z que irão zerar ou estão zerados no estoque e P1, P2 e P3 são os pesos para cada variável respectiva.

A média ponderada de Materiais Excedentes (MME) tem-se:

$$MME = \frac{(QtdeMatExcedentesX * P1 + QtdeMatExcedentesY * P2 + QtdeMatExcedentesZ * P3)}{(P1 + P2 + P3)} \quad (7)$$

onde $QtdeMatExcedentesX$ é a quantidade de materiais de classificação X que estão excedentes no estoque, $QtdeMatExcedentesY$ representa a quantidade de materiais de classificação Y que estão excedentes no estoque, $QtdeMatExcedentesZ$ é a quantidade de materiais de classificação Z que estão excedentes no estoque e P1, P2 e P3 representam os pesos para cada variável respectiva.

A quantidade de materiais de um determinado parâmetro medido foi separada em quantidades organizadas, mediante a sua *classificação XYZ*, tornando-se possível acrescentar pesos para cada tipo de classificação. Por exemplo, um material do tipo Z possui maior importância operacional e maior peso no estoque do que um material do tipo Y que, por sua vez, possui um peso maior do que um do tipo X. Os pesos utilizados nos cálculos dos percentuais foram: P1 = 1, P2 = 2 e P3 = 7. Maiores informações sobre a classificação de materiais XYZ, podem ser obtidas no anexo B.

Para o cálculo da média do EGE também foi adotado um sistema de média ponderada, pois foi necessário acrescentar pesos nos parâmetros analisados. Desta forma, os parâmetros que ficaram com maior peso foram respectivamente: MMZ, MMLVV e MME. Os *Materiais Zerados* possuem um maior peso no estoque, pois definem o valor vitalício para o estoque. Os *Materiais com Lotes em Véspera de Vencimento* possuem um peso menor do que a falta de materiais, mas não deixam de ter certa importância no estoque. Já os *Materiais Excedentes* possuem menos peso, o que não deixa de ser uma situação

desconfortável para um setor. No entanto, é uma situação menos agravante em relação aos outros parâmetros discutidos.

O cálculo da MEGE (Média do Estado Geral do Estoque) é apresentado pela fórmula abaixo.

$$MEGE = \frac{(PME * P1 + PLVV * P2 + PMZ * P3)}{(P1 + P2 + P3)} \quad (8)$$

Cálculo do Percentual do Nível de Risco

Os valores dos parâmetros do EGE, obtidos nos cálculos de média ponderada, foram convertidos em percentual, conhecido como *Nível de Risco* (NR), para realizar uma normalização dos dados. Através do percentual de cada parâmetro, pode-se quantificar a sua gravidade, tornando possível o AGT interpretá-lo, a fim de tomar as estratégias para a sua resolução. O grau de Nível de Risco está disposto na Tabela 3, conhecida como *Tabela de Níveis de Risco*.

Percentual (%)	Estado do Parâmetro
0,0 - 33,33	Bom
33,4 - 66,66	Razoável
66,7 - 100	Crítico

Tabela 3 - Tabela de Níveis de Risco.

O NR de cada parâmetro do EGE foi calculado embasado no cálculo de percentual, disposto pela seguinte fórmula:

$$NR = \frac{(MediaPonderadaDoParametro * 100\%)}{MediaPonderadaSituacaoPiorCasoDoEstoque} \quad (9),$$

onde Nível de Risco (*NR*) é o resultado da multiplicação entre média ponderada dos parâmetros (*MediaPonderadaDoParametro*: MMZ, MME, MMLVV e MEGE) por cem por cento, e, a divisão pela média ponderada da situação do estoque no pior caso (*MediaPonderadaSituacaoPiorCasoDoEstoque*).

A *MediaPonderadaSituacaoPiorCasoDoEstoque* foi obtida com base nos conceitos de classificação de material XYZ (para maiores detalhes, vide anexo B). Foi feito um levantamento sobre o pior caso que poderia ocorrer no estoque, que é a lista de materiais ser composta apenas por materiais do tipo Z, o qual representa o tipo de material imprescindível para a sobrevivência de um setor e detém o maior peso no cálculo.

Os NR dos parâmetros calculados do EGE são armazenados na base de dados do AGT e é acessado por ele para analisar sobre tais dados e tomar as estratégias para normalizar o estoque.

4.7 Síntese do Capítulo

Este capítulo abordou a construção de um modelo de sociedade de STIs, destinado a trabalhar na vertente da saúde. Esta é uma proposta para a solução da má distribuição dos materiais, por meio de estratégias que fazem o tratamento da lista de itens que estão em estado irregular nas UBSs de uma rede municipal de saúde.

Foi apresentada a arquitetura de um AGT, modelada sobre a arquitetura de um STI, bem como a suas funções, funcionamento e sua organização social. Também foi enfatizado neste capítulo sobre a estrutura do AGT, organizada em camadas: cognitiva, controle e comunicação. E por fim, foi abordado sobre as estratégias de resolução do problema, baseadas nos conceitos de gestão de estoque.

4.8 Bibliografia

- [ALV1997] Alvares, L. O., Sichman, J. S. *Introdução aos Sistemas Multiagentes*. In: Medeiros, Cláudia Maria Bauzer (Ed.). *Jornada de Atualização em Informática (JAI'97)*. Brasília: UnB, 1997. cap. 1, p. 1–38.
- [BOR2003] Bordini, R. H. e Vieira, R. *Linguagens de programação orientadas a agentes: uma introdução baseada em AgentSpeak(L)*. *Revista de Informática Teórica e Aplicada*, X(1):7–38. Instituto de Informática da UFRGS, Brazil, 2003.
- [BOR2005] Bordini R. H.; Hübner J. F.; *et al.* *Jason - A Java-based agentSpeak interpreter used with saci for multi-agent distribution over the net*. Department of Computer Science University of Durham Durham DH1 3LE, U.K, 2005.
- [FIN1993] Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, F., McKay, D., McGuire, J., Pelavin, P., Shapiro, S., *Specification of the KQML Agent-Communication Language*, Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, Updated in July, 1993.
- [GEN1994] Genesereth, M.R., Ketchpel, S. P. *Software Agents*. *Communications of the ACM*. Vol. 37 (7). July 1994, p48-53.
- [HUB2004] Hübner, J. F., Bordini, R. H., Vieira, R. *Introdução ao desenvolvimento de Sistemas Multiagentes com Jason*. Departamento de Sistemas e Computação, Universidade Regional de Blumenau (FURB), 2004. p.1-12.
- [HUB2003a] Hübner, J. F., Sichman, J. S. *Saci Programming Guide*, USP - Universidade de São Paulo, São Paulo, jul 2003.

[HUB2003b] Hübner, J. F., Sichman, J. S. SACI: *Uma Ferramenta para Implementação e Monitoração da Comunicação entre Agentes*, LTI - Laboratório de Técnicas Inteligentes, USP - Escola Politécnica da Universidade de São Paulo, 2003, Artigo.

[HUH1999] Huhns, M. N., Larry M. S. *Multiagent Systems and Society of Agents*. 1999, 96p

[MAT2006] Mattos, E. P., Mendes L. S., Bottoli M. *Uma Proposta de Sociedade de Agentes Inteligentes para a Gestão de Medicamentos nas Unidades Básicas de Saúde*. CBIS - X Congresso Brasileiro de Informática em Saúde "Informática em Saúde e Cidadania". Florianópolis – SC. 14 a 18 de outubro de 2006 – Artigo.

[SHO1993] SHOHAM, Y. *Agent-oriented programming*. Artificial Intelligence, Amsterdam, v.60, p.51-92, 1993.

[WEI1999] Weiss, G. *Multyagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 1999, p. 121 Ed. Gerhard Weiss.

5 Resultados Obtidos

5.1 Introdução

Este capítulo tem por objetivo apresentar os resultados referentes à simulação da sociedade de agentes AGTs, durante seis meses de cinco setores distintos. Deve-se considerar que cada setor recebeu uma lista de materiais em comum, porém, com as suas respectivas cotas.

Conforme apresentado no capítulo anterior, diariamente, cada AGT representante de cada UBS realizou os cálculos de gestão e de estado geral do estoque. Adotou estratégias pedagógicas, com o propósito de orientar o usuário a tratar o estado geral do estoque.

O capítulo presente está dividido em duas seções. A primeira trata do comparativo entre o estoque real e o ideal dos setores, enfatizando a sua convergência ocorrida pela repercussão causada pelo AGT, em cada UBS. Será apresentada a análise dos resultados obtidos de três dos cinco itens da simulação de cada setor, sendo que os demais encontram-se no anexo A. Os gráficos estão dispostos por item para serem comparados mais facilmente.

A segunda seção contém a análise do modelo do aprendiz (ou estado geral do estoque). Ela ilustra o seu comportamento e atitudes tomadas, com o auxílio do AGT, perante as mais variadas situações encontradas no decorrer da simulação.

5.2 Estrutura da Simulação

Como obtenção dos resultados foi montada em laboratório uma estrutura similar à descrita anteriormente, onde um almoxarifado central (almoxarifado gestor) é responsável por distribuir materiais e medicamentos aos seus setores de consumo, conhecidos como UBSs. Foi implantado uma rede de distribuição constituída por cinco setores distintos, cada setor é representado por uma instância cliente do Sig2m com dados autênticos⁹, supridos

⁹ Bases de dados de setores reais com pelo menos dois anos de uso, fornecido pela prefeitura municipal de Campinas.

mensalmente pelo almoxarifado central. O Almoxarifado central é representado por um computador que contém as instalações *Master* e *Server* do Sig2m, com o propósito de gerar as operações de expedição e recebimento de requisições de materiais das UBSs. O período de duração da simulação foi de seis meses. Para realizar esta análise, foi selecionada uma lista de materiais composta por cinco itens distintos, comumente movimentados nos setores, alguns com maior movimentação em alguns setores do que em outros. Os critérios de seleção dos itens da lista foram:

- Estimativa entre os itens mais movimentados: Quanto maior a movimentação dos itens, mais precisa seria os cálculos de gestão de estoque. Pois os cálculos de gestão dependem diretamente das movimentações de entrada e saída de tais itens.
- Lista composta por materiais de diferentes tipos de classificação: A lista deveria ser composta por itens que variam de níveis de importância no estoque, ou seja, itens de classificação entre X e Z.
- Itens que possuem lotes em véspera de vencimento: Tentativa de testar a capacidade do agente analisar e quantificar o estado dos itens que possuem lotes em véspera de vencimento, bem como tomar as devidas decisões sobre o mesmo, a fim de saná-lo.

O objetivo da simulação foi:

- Verificar a viabilidade do AGT de cada setor no auxílio ao usuário a gerir o seu estoque;
- Verificar a viabilidade do protocolo de comunicação implementado para a negociação das requisições / doações de material;
- Averiguar a eficiência da estrutura de planos BDI, implementada para o AGT na linguagem *AgentSpeak(L)*;
- Verificar os impactos da sociedade de tutores inteligentes na gestão do estoque distribuído intersetores;
- Verificar os impactos da sociedade de tutores inteligentes na gestão do estoque do almoxarifado central.

5.3 Síntese do processo de distribuição de materiais e medicamentos para os setores da rede municipal de saúde

De forma genérica, todos os setores da sociedade seguem o mesmo conjunto de processos, conforme ilustrado pela Figura 47: Inicialmente, ocorre a entrada de materiais nos setores (passo 1 da Figura 47), onde o agente realiza os cálculos de gestão de estoque, gerando o estoque ideal do item (passos 2 e 3). Através do comparativo do saldo ideal é obtida a lista de materiais excedente e zerados (passo 4). O AGT realiza uma análise sobre a lista e adota a estratégia de informar ao usuário sobre a situação do estoque. Seqüencialmente, é realizado o processo de requisição / doação de materiais intersetores (passo 5). Em seguida, é feito um acerto do estoque ideal e real através da devolução dos itens excedentes para o almoxarifado gestor, passo 6 da Figura 47. Finalmente em véspera do término de cada ciclo, é realizado o cálculo da cota ideal do item e enviado para o almoxarifado central, passo 7 da Figura 47. Esta foi a estratégia genérica de processos seguida por cada setor da sociedade, na tentativa de balancear o estoque distribuído de materiais intersetores.

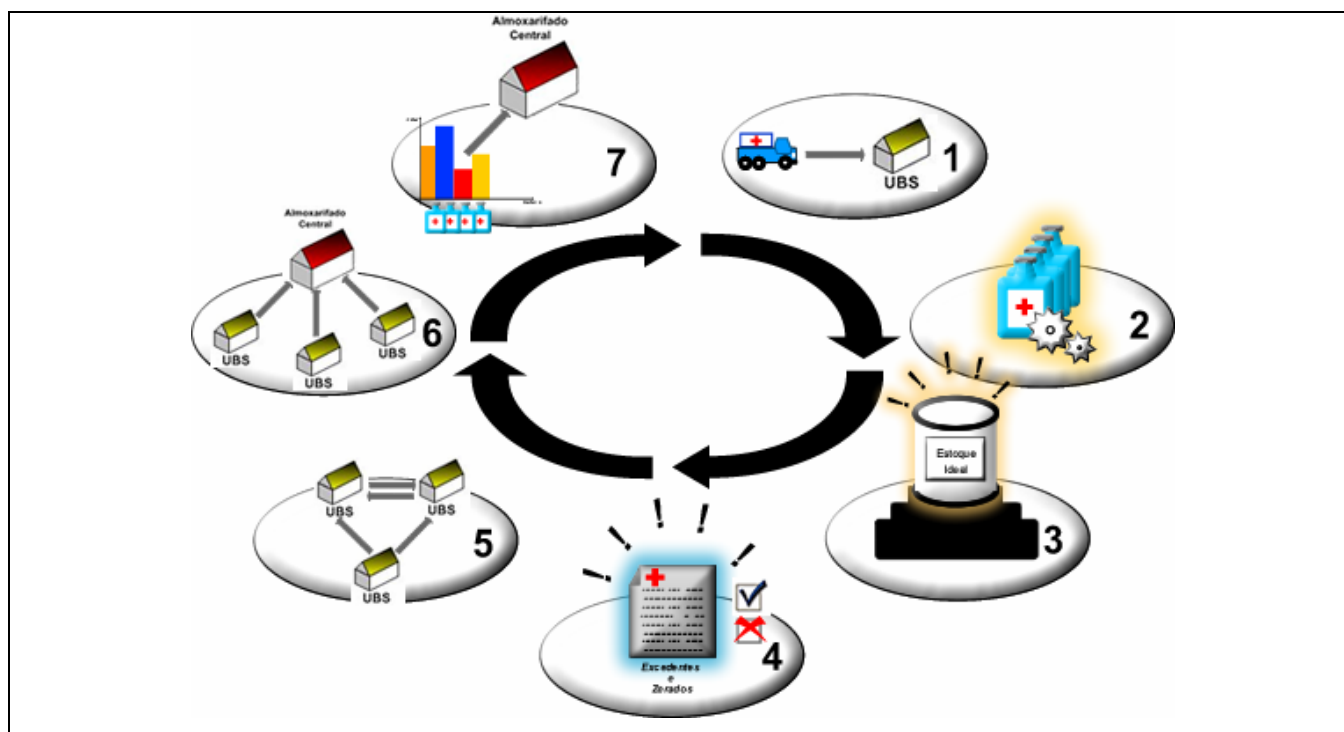


Figura 47 - Ciclo de distribuição de materiais e medicamentos na rede municipal de saúde.

5.4 Notas sobre a simulação

Os resultados da simulação rendeu para cada setor um total de seis gráficos: um gráfico para cada item da lista de suprimento, a qual é enviada mensalmente pelo almoxarifado gestor. Estes gráficos representam as curvas do estoque real e ideal de cada item e um gráfico da coreografia dos parâmetros do Estado Geral do Estoque (EGE), no decorrer dos seis meses de processamento.

5.5 Resultados da coreografia do histórico do estoque real

Esta seção apresenta a coreografia do estoque de três matérias nas UBSs da rede municipal. Os itens aqui apresentados possuem a classificação do tipo Z, cuja falta causa sérios danos operacionais no estoque. No contexto em questão, tratam-se de medicamentos cuja ausência pode causar sérios problemas ao paciente (no caso de o paciente necessitar de medicamentos controlados). Por esta razão, foram selecionados itens do tipo Z, que são os principais elementos influentes nas alterações da situação do estado do estoque, levando em consideração a realidade das redes municipais de saúde.

5.5.1 Análise dos Resultados do Material 1 nas Unidades Básicas de Saúde

A partir da data do dia 15 de Agosto de 2006, cada AGT da sociedade começa a atuar no seu respectivo setor, no ato do recebimento da lista de suprimento de itens. Em tese, a lista varia apenas nas cotas de materiais que são enviadas para os setores.

A estratégia inicial tomada pelo AGT foi obter a lista de itens que estão excedentes e zerados, a partir dos cálculos de gestão que foram calculados com base no histórico de consumo de cada material. Com a lista de excedentes, o AGT estimou um saldo ideal do dia, suficiente a atender as necessidades do setor até a chegada de materiais do ciclo seguinte. O cálculo do saldo ideal foi feito pelo AGT, na tentativa de nortear o usuário sobre o desvio entre a curva do saldo real e o saldo ideal. Este comportamento pode ser notado na Figura

48, no qual é encontrado duas curvas, uma que representa o saldo real do estoque (linha continua) e a outra que representa o saldo ideal (linha pontilhada).

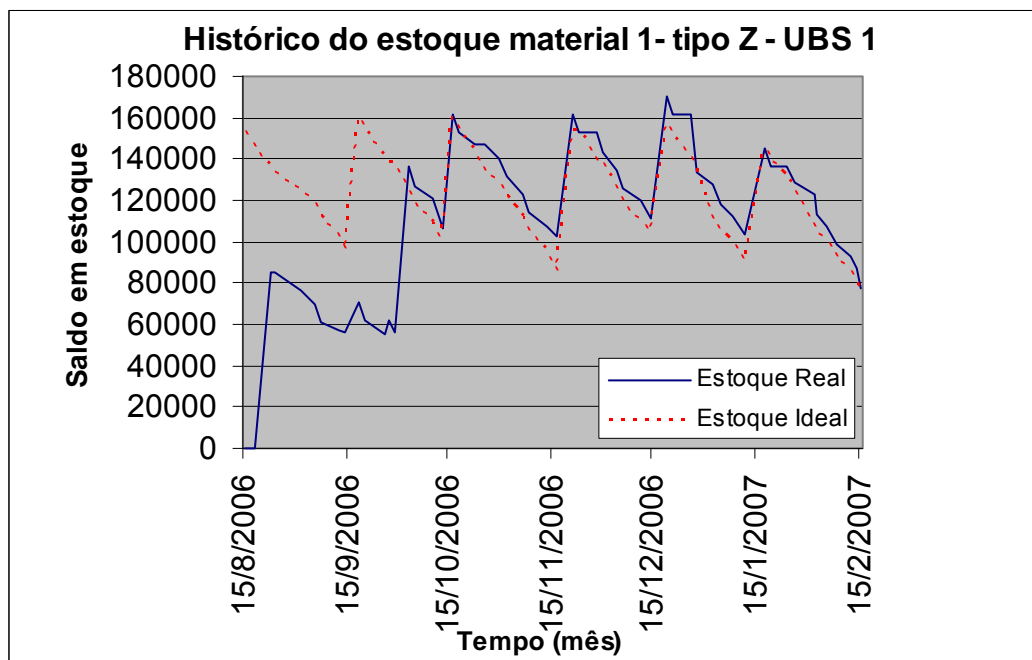


Figura 48 - histórico do estoque do material 1, UBS 1.

Na data 15/08/2006, ilustrado na Figura 48, o AGT alertou o usuário sobre a divergência entre o saldo do estoque real e o saldo ideal, indicando a quantidade de material necessária para que o seu estoque fosse normalizado. O saldo real estava zerado e em alguns dias o seu estado passou a ser 80000. Esta repentina subida foi causada pelas doações geradas pelas UBS 2, UBS 3 e UBS 5 à UBS 1 que, por sua vez, requisitou materiais através da sociedade de AGTs.

As UBSs que doaram material possuíam o item em estado excedente, ou seja, seu estoque em estado irregular. Desta forma, cada setor tomou a medida preventiva de doar o material 1 para a UBS 1. Isto pode ser denotado como um pequeno declive, logo após a data 15/08/2006 para as seguintes figuras: Figura 49, Figura 50 e Figura 52, que representam respectivamente as UBSs 2, 3 e 5. Entretanto, as doações não foram suficientes para atender as necessidades do estoque da UBS 1, mas sim como uma medida paliativa para manter um estado próximo ao ideal, até que seja tomada uma providência definitiva, como por exemplo, o suprimento pelo almoxarifado central.

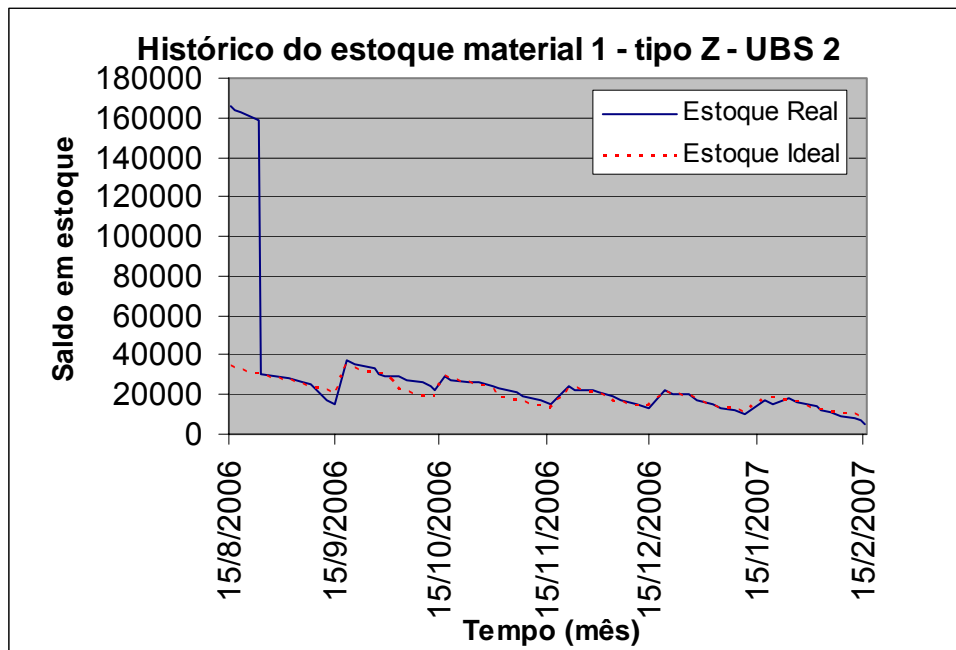


Figura 49- histórico do estoque do material 1, UBS 2.

Outro fator importante a ser notado, entre as datas de 15/08/2006 e 15/09/2006, para os gráficos representados pelas figuras: Figura 49, Figura 50 e Figura 52, ocorreu uma queda brusca do estoque real, ou seja, foi realizada uma movimentação de saída de uma quantidade significativa deste item. Tal fato é justificado pela quantidade excedente desnecessária do material 1, que foi expedida para o almoxarifado gestor. Esta medida foi tomada pelo usuário, através das orientações feitas pelo AGT para normalizar o saldo do estoque, ou seja, o saldo real tangenciar um saldo ideal, na tentativa de sanar a quantidade excedente do item.

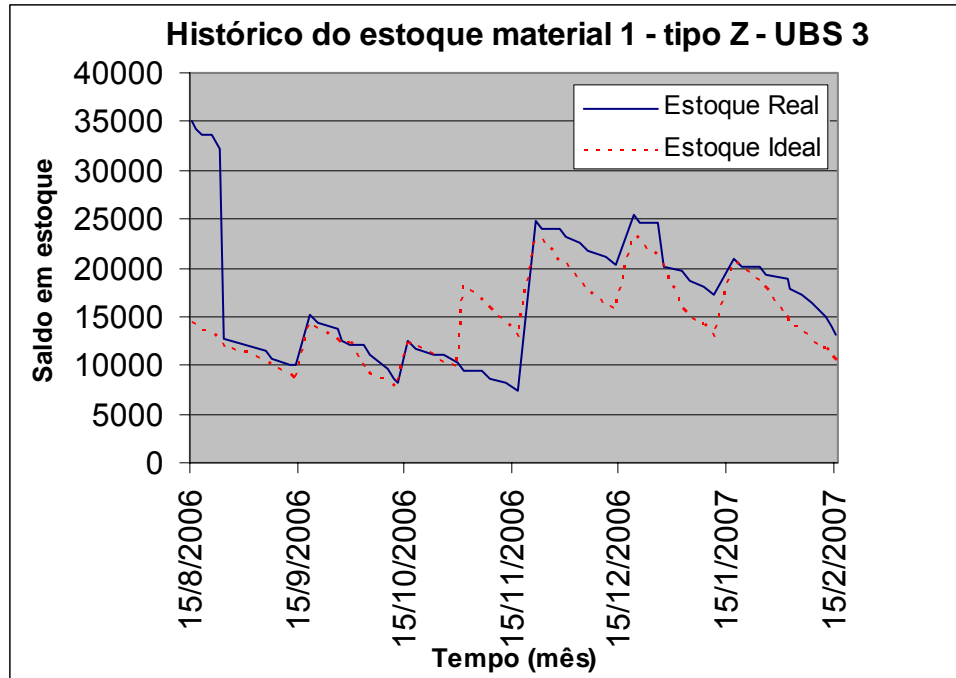


Figura 50- histórico do estoque do material 1, UBS 3.

No início do primeiro mês, a UBS 4 se encontrava em situação análoga à encontrada na UBS 1. O seu saldo real estava zerado e com grande divergência em relação ao saldo ideal. Através das orientações do AGT, a UBS 4 tomou as providências de tratamento do estoque, requisitando o material 1 para os setores da sociedade que, por sua vez, o atendeu. Este fato é notado na Figura 51, no qual ocorreu a mudança repentina no saldo real, de valor zero para 120, na data de 18/08/2006.

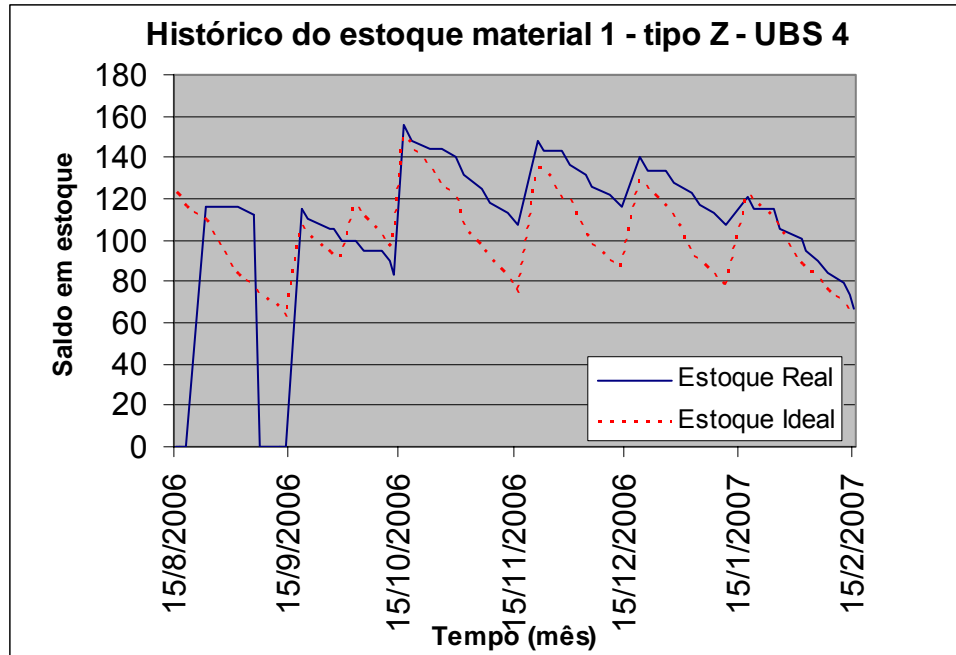


Figura 51 - histórico do estoque do material 1, UBS 4.

Próximo à data 15/09/2006, data de suprimento, o material 1 esgotou antes de seu reabastecimento. Esta queda ocorreu por dois motivos: a pequena quantidade em estoque do item, o que causou a variação do gráfico; e por um consumo irregular próximo à data de suprimento. Antes do dia 15 de cada mês (data de suprimento), o AGT realizou os cálculos da cota ideal de cada item e o enviou ao almoxarifado gestor. O almoxarifado gestor, por sua vez, realizou o processo de suprimento dos setores. Para cada setor, o almoxarifado analisou as novas listas de cotas dos materiais, enviadas por cada um, constituindo a lista de suprimento do ciclo seguinte.

No segundo mês de gestão (15/09/2006), o almoxarifado gestor fez a distribuição materiais para os setores. Para a maioria dos setores, o saldo real convergiu absolutamente para o ideal, com exceção da UBS 1, cuja convergência entre as duas curvas passou a ser maior nos meses seguintes. De um modo geral, todos os setores passaram a ter uma convergência maior entre o estoque real e o ideal no terceiro mês, devido ao acerto realizado no estoque nos meses anteriores, deixando cada setor com um perfil de estoque coerente às suas necessidades, através das cotas ideais calculadas pelo AGT e as orientações diárias acerca de seu estado geral ao usuário.

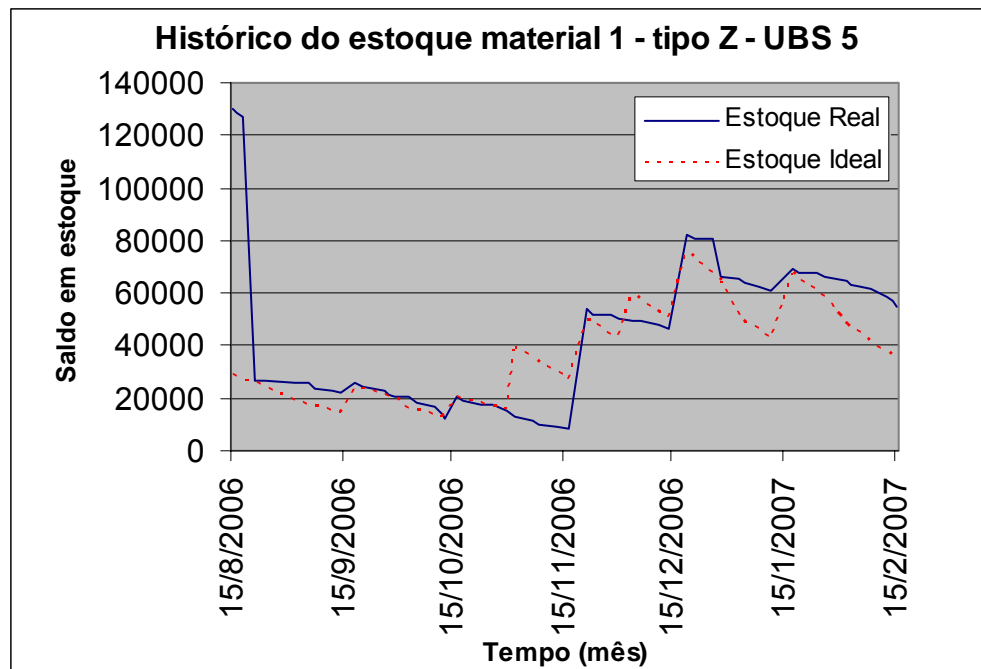


Figura 52 - histórico do estoque do material 1, UBS 5.

5.5.2 Análise dos Resultados do Material 2 nas Unidades Básicas de Saúde

No início da simulação, o material 2 estava em falta na UBS 1, conforme mostrado na data 15/08/2006, na Figura 53. Da mesma forma que no tópico anterior, o AGT alertou o usuário acerca do estado do estoque que, por sua vez, requisitou os materiais em falta para a sociedade de tutores. Entretanto, o atendimento não foi feito por completo. Foi doado para a UBS 1 apenas um montante de 44000, mas que abrandou o problema até a data de suprimento.

Em todo o tempo de simulação, a UBS 1 apresentou grande quantidade do material 2 em estoque não correspondente ao seu consumo. Isto é justificado devido ao histórico de consumo do item ter sido consideravelmente significativo nos meses anteriores, o que repercutiu nos meses da simulação. No entanto, ao observar a curva do estoque ideal na Figura 53, é denotada, em toda a simulação, a tendência de redução do consumo, prevista para os próximos meses.

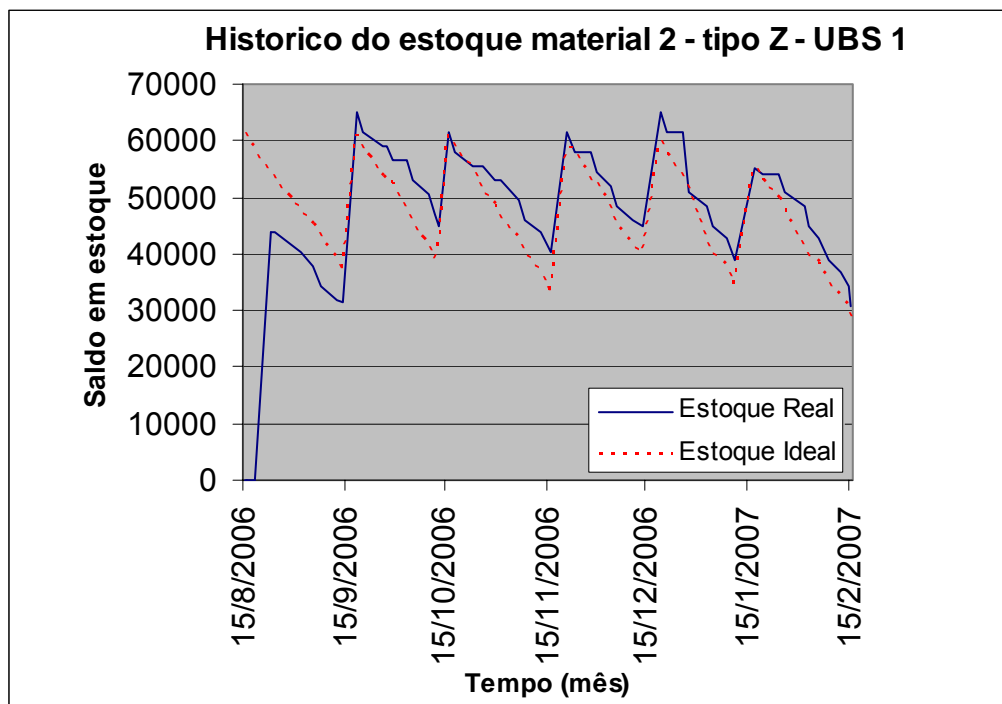


Figura 53 - histórico do estoque do material 2, UBS 1.

As UBSs que atenderam à requisição da UBS 1 foram a UBS 2, UBS 4 e UBS 5, de forma que este atendimento apresenta um ligeiro declive em seu estoque como é mostrado próximo a data de 15/08/2006 nas figuras: Figura 54, Figura 56 e Figura 57.

Por volta do dia 20/08/2006, é feito o tratamento do item excedente nos setores. É realizada a devolução dos materiais excedentes para o *almoxarifado gestor*, mostrado nas figuras: Figura 54, Figura 56 e Figura 57, nos quais é encontrado uma queda no estoque real, se aproximando do ideal.

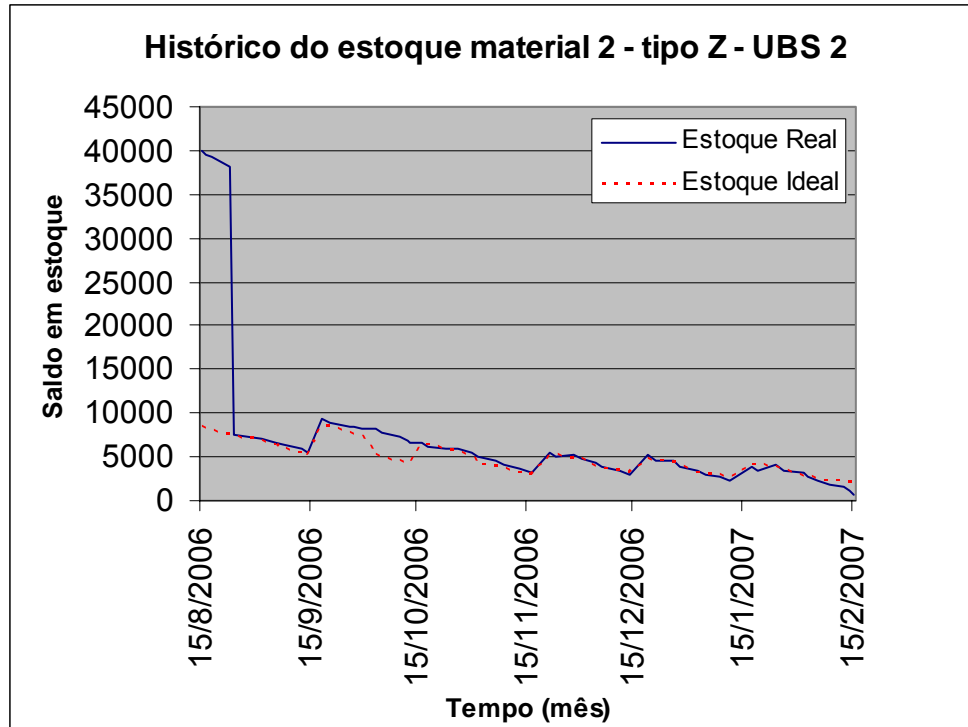


Figura 54 - histórico do estoque do material 2, UBS 2.

No mês dois, a tendência do estoque real para o ideal passa a ser mais acentuada. O estoque passa a atingir um padrão ideal, ou seja, um equilíbrio entre a demanda e o consumo. Neste período, não houve doações entre as UBSs, pois não foi registrado algum indício de falta de materiais. Cada UBS consumiu o seu próprio estoque e, na véspera do fim do ciclo, o AGT realizou os cálculos da cota ideal dos itens, a ser enviada para o almoxarifado gestor para o suprimento para o mês seguinte.

No terceiro mês (entre 15/10/2006 e 15/11/2006), a UBS 3 fez a requisição do item 2 para os setores, mas apenas a UBS 1 a atendeu, como ilustrado na Figura 55.

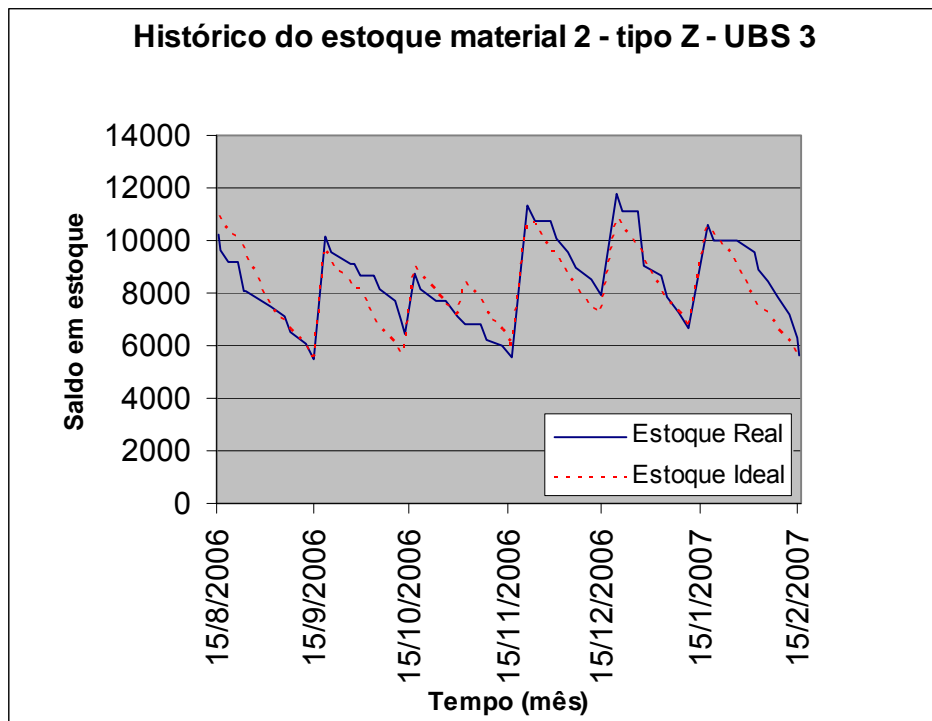


Figura 55 - histórico do estoque do material 2, UBS 3.

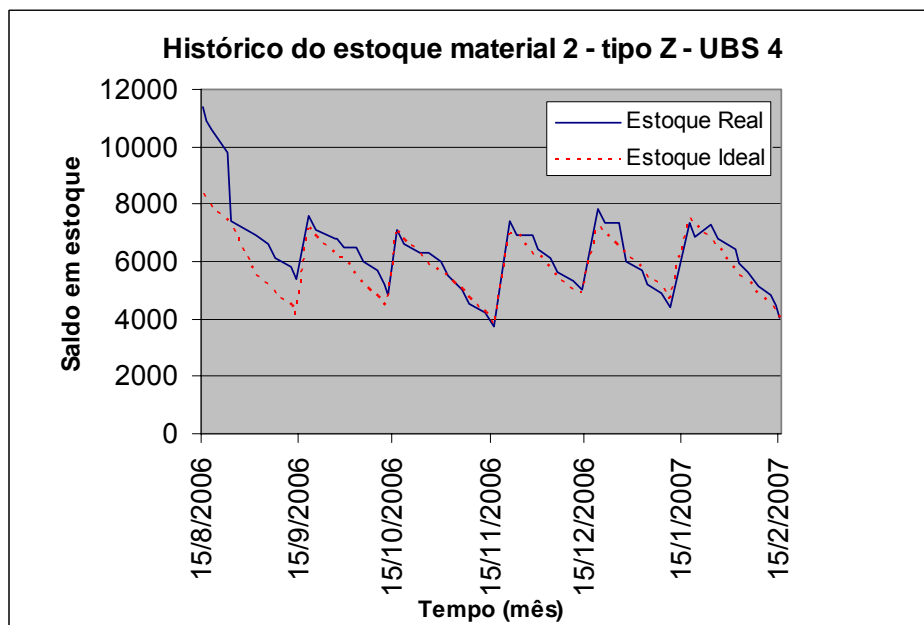


Figura 56 - histórico do estoque do material 2, UBS 4.

No quinto mês (15/12/2006 e 15/01/2007), apenas a UBS 5 realizou a devolução das quantidades excedentes para o almoxarifado gestor. Isto é indicado por uma pequena queda do estoque na Figura 57, por volta do dia 20/12/2006.

Um outro ponto a ser notado na UBS 5, é o aumento do estoque ideal no decorrer do tempo da simulação. Este crescimento do estoque ideal se dá ao fato do aumento do consumo do material 2 durante a simulação, o que conseqüentemente influi no estoque ideal. Isto implica que o estoque ideal está intimamente ligado ao processo de gestão de materiais através da apresentação de valores que acompanham as tendências de previsão de consumo do estoque.

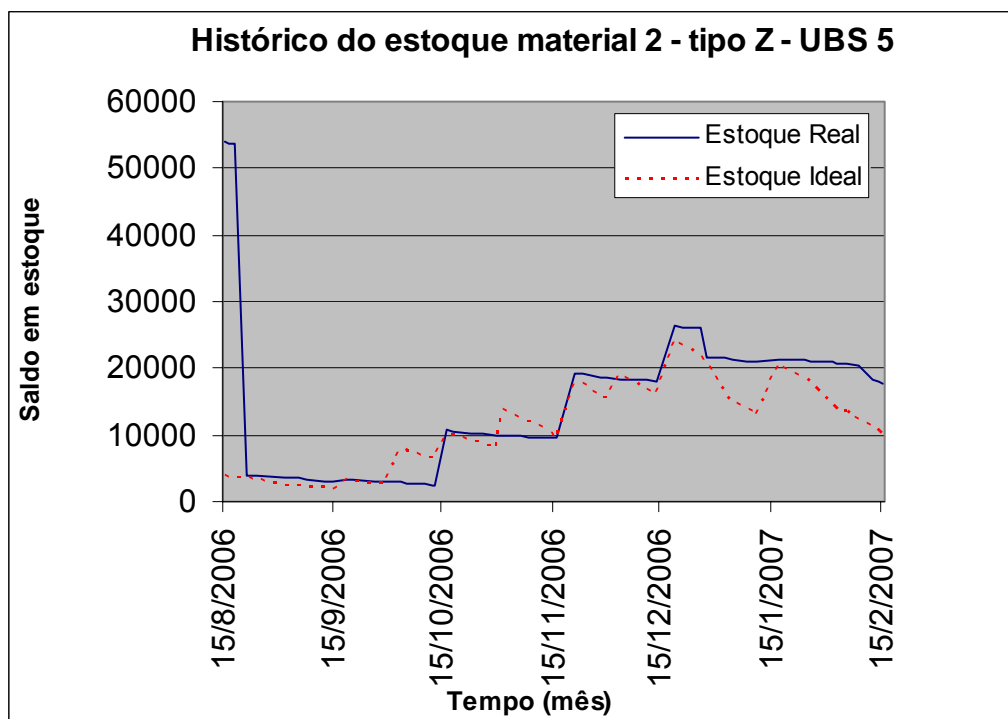


Figura 57 - histórico do estoque do material 2, UBS 5.

De um modo geral, a partir do segundo mês o estoque real do material 2 passou a convergir para o ideal, mantendo esta tendência para os outros demais meses.

5.5.3 Análise dos Resultados do Material 3 nas Unidades Básicas de Saúde

O material 3 assumiu o mesmo comportamento que os demais citados nas seções anteriores. Alguns setores possuíam o material 3 em saldo excedente e outros, em saldo zerado. Para a sua regularização, foram adotados os mesmos procedimentos que os dos outros materiais. Na Figura 58, o estoque do material 3 estava em estado excedente, disponível para gerar doações para os demais setores, como é o caso da UBS 2 (Figura 59), em que este item estava com saldo zerado. Diante de tal fato, a UBS 2 realizou o processo de requisição para todos os setores da sociedade, na espera de uma proposta de doação.

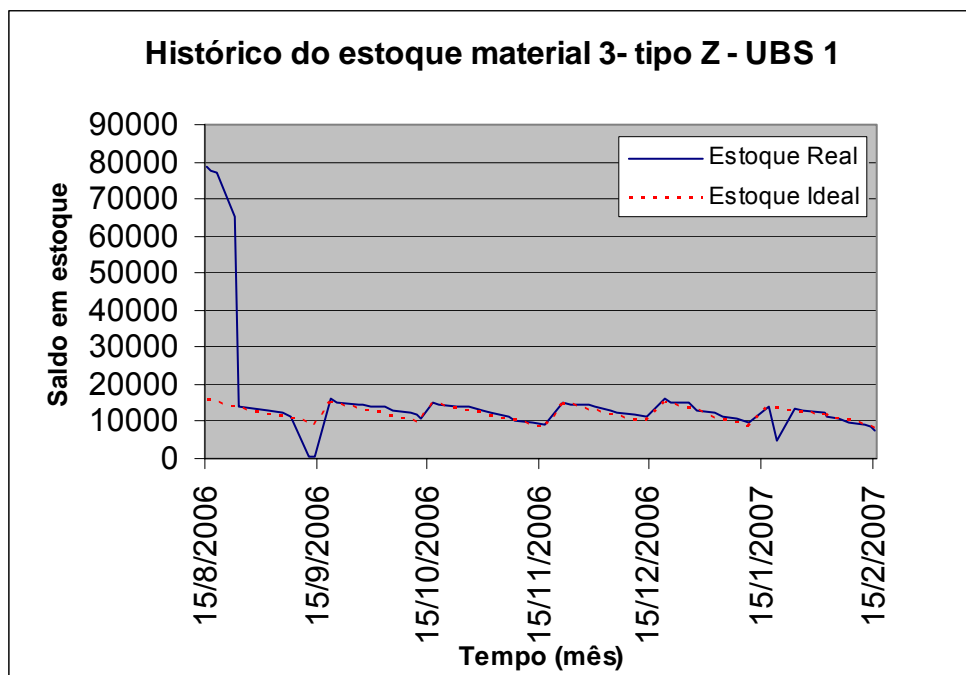


Figura 58 - histórico do estoque do material 3, UBS 1.

Chegada a lista de doações, vinda dos setores candidatos à doação, a UBS 2 confirmou a aceitação da proposta de doação da UBS 1 que, por sua vez, realizou a saída do material para o seu destinatário, indicado como um pequeno declive na Figura 58, próximo à data do dia 15/08/2006.

Ao fim do ciclo do primeiro mês, ocorreu um fato inesperado no estoque dos materiais (vide Figura 58). Repentinamente, o material 3 zerou próximo à data de suprimento. Uma

situação incomum, pois não se trata de um período sazonal¹⁰, mas sim de um repentino consumo do item. No entanto, a sua reposição foi feita em imediato, através do abastecimento mensal, realizado pelo almoxarifado.

Na Figura 58, pode ser facilmente visualizado a entrada de materiais, fruto da doação da UBS 1, indicado como um aumento do saldo de 0 para 9000.

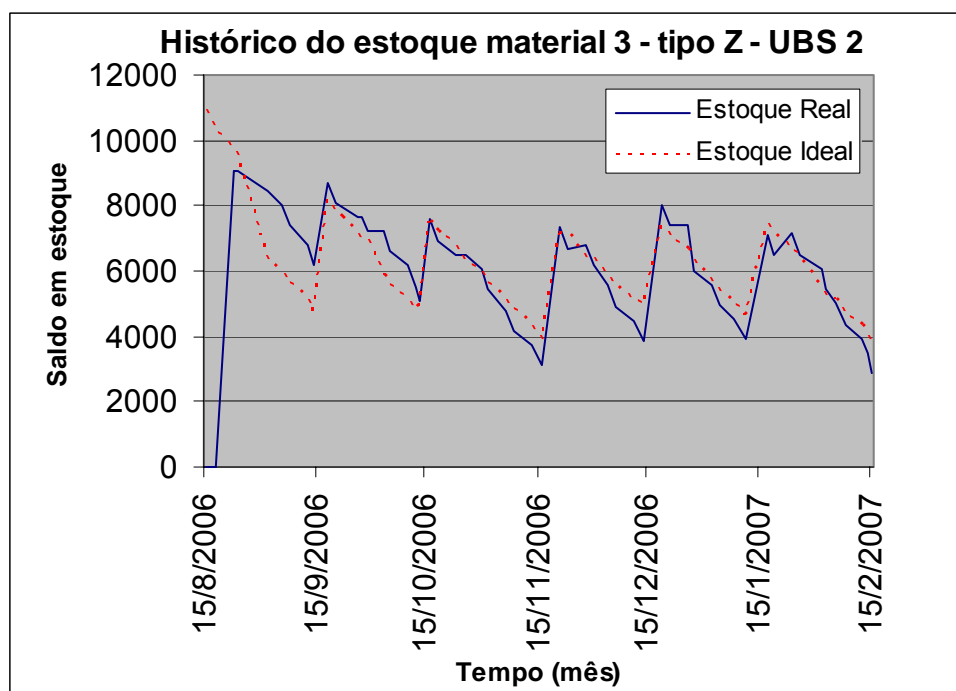


Figura 59 - histórico do estoque do material 3, UBS 2.

Já na UBS 3 (representado na Figura 60), apesar da curva de estoque real estar abaixo do seu estoque ideal, este optou por consumir o seu estoque, sem realizar requisição para nenhum setor, ultrapassando a margem de segurança (ou estoque de segurança). Uma atitude considerada inadequada, quando se trata de um item que possui uma classificação Z, material de importância vital para a UBS.

¹⁰ Informações sobre sazonalidade de material, verificar o anexo B deste trabalho.

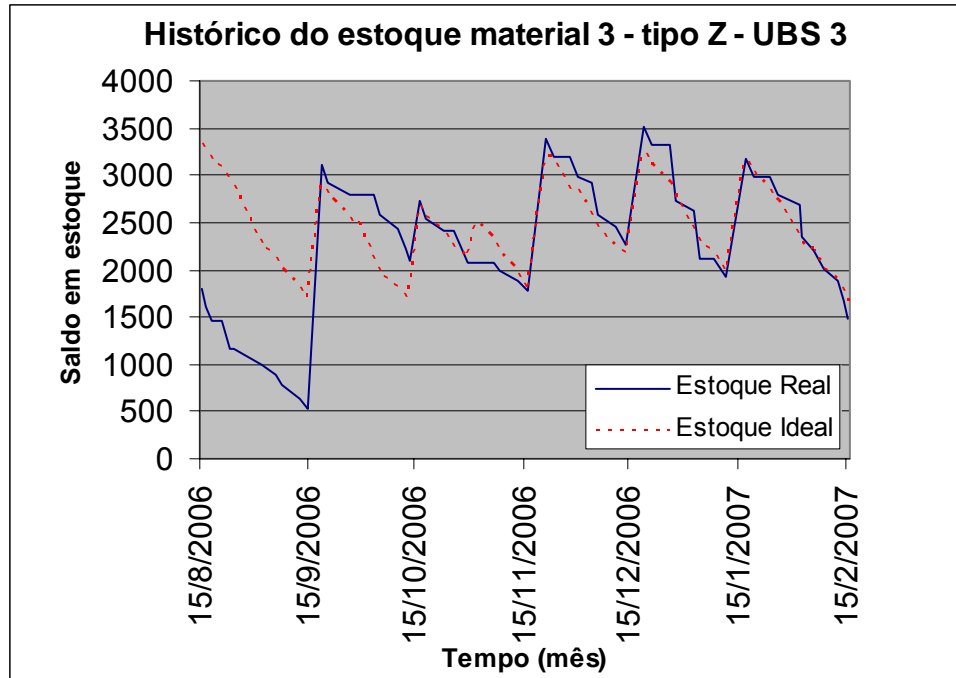


Figura 60 - histórico do estoque do material 3, UBS 3.

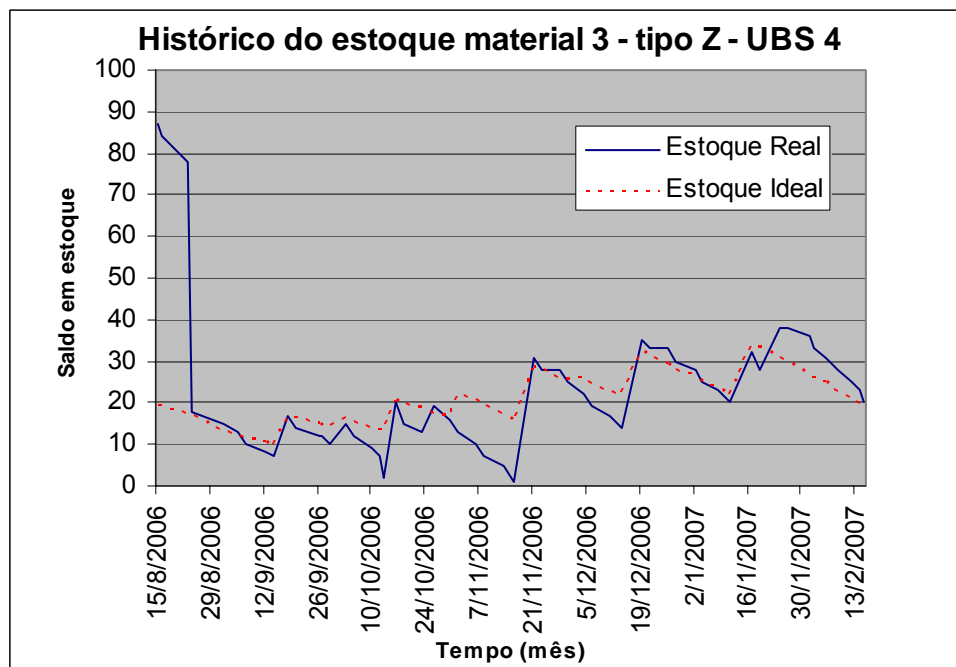


Figura 61 - histórico do estoque do material 3, UBS 4.

Próximo à data do dia 20/08/2006, as UBSs 1, 3 e 4 realizaram a devolução dos itens excedentes para o almoxarifado gestor, a fim de regularizar o estoque e também o envio da lista de cotas, ora calculada pelo AGT.

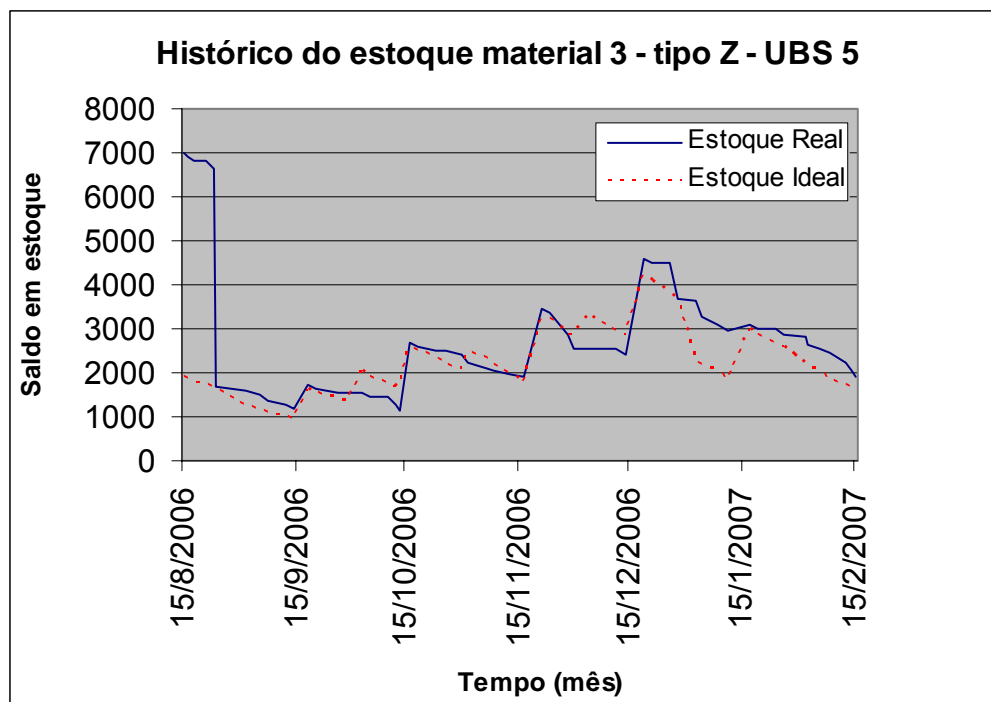


Figura 62 - histórico do estoque do material 3, UBS 5.

No segundo mês, o almoxarifado gestor enviou as cotas corrigidas em conformidade com as necessidades de cada setor, ilustrado na data de 15/09/2006. No decorrer do segundo mês não houve requisições e ou doações entre os setores, tão pouco devoluções para o almoxarifado gestor. O saldo real manteve a convergência com o saldo ideal. De um modo geral, para os setores, os meses assumiram o mesmo perfil que o do segundo mês, com exceção ao terceiro mês na UBS 4, denotado pela Figura 62, em que houve um consumo repentino, que chegou a atingir o estoque de segurança.

5.6 Conclusão da Análise do Estoque real e Ideal

Como pôde ser notado no decorrer da simulação, todos os ciclos seguem um mesmo perfil, uma seqüência padrão de processos que são executados de forma seqüencial. Inicialmente é realizado o processo de requisição e doação entre os setores, na tentativa de sanar os itens que estão em estado zerado nos setores. Em seguida, ocorre o processo de devolução dos itens para o almoxarifado gestor que não foram doados, na tentativa de manter o estoque em coerência com o estoque ideal, o que foi o ponto crucial para o balanceamento do estoque. Esta foi a estratégia adotada como padrão, para a resolução do processo de balanceamento de carga do estoque distribuído.

Um outro ponto em questão é que a coreografia do histórico do estoque, nos meses subseqüentes, passou a repercutir de forma acentuada no terceiro mês, ponto de sincronia em que os estoques real e ideal tendem a uma convergência, conforme ilustrado nos gráficos apresentados neste capítulo e em anexo.

A freqüência de requisições tanto para os setores quanto para o almoxarifado gestor tendeu a reduzir no decorrer dos meses, limitando apenas ao complemento do suprimento mensal, extinguindo a falta de medicamentos nos setores.

5.7 Análise da coreografia dos Parâmetros do Estado Geral do Estoque (EGE)

5.7.1 Introdução

Nesta seção, serão apresentados os resultados referentes ao modelo do aprendiz, camada da arquitetura de um STI, que tem por finalidade armazenar as ações do usuário perante o seu estoque.

Com a lista de materiais excedentes e zerados, o AGT calculou três parâmetros do EGE, apresentados no capítulo 4, compreendidos em: percentual de materiais excedentes (PME), percentual de materiais zerados (PMZ) e o percentual de materiais com lotes em véspera de vencimento (PMLVV). Diante de tais informações, tornou-se possível quantificar e qualificar o

estado geral do estoque, de modo que o AGT o utilizou como referência para regularizar a situação do estoque através de orientações ao usuário sob a forma de mensagens.

5.7.2 Análise dos parâmetros do EGE da UBS 1

A Figura 63 apresenta o modelo do aprendiz, que encontra-se disposto sob a forma dos três parâmetros percentuais (PMZ, PME e PMLVV) e também em um percentual geral calculado a partir dos três parâmetros, conhecido como *Percentual do Estado Geral do Estoque* (PEGE). O gráfico contém informações de cada parâmetro avaliado durante os seis meses de simulação.

No início do primeiro ciclo (mês), o PEGE se encontra em estado razoável, em virtude de alguns itens da lista que estão zerados no estoque, apresentado na Figura 63. Quando a quantidade de itens zerados aumenta a partir da segunda metade do mês, o percentual de materiais zerados (PMZ) atinge um estado crítico próximo à data de suprimento, devido ao processo de convergência do estado real para o ideal para a lista de materiais.

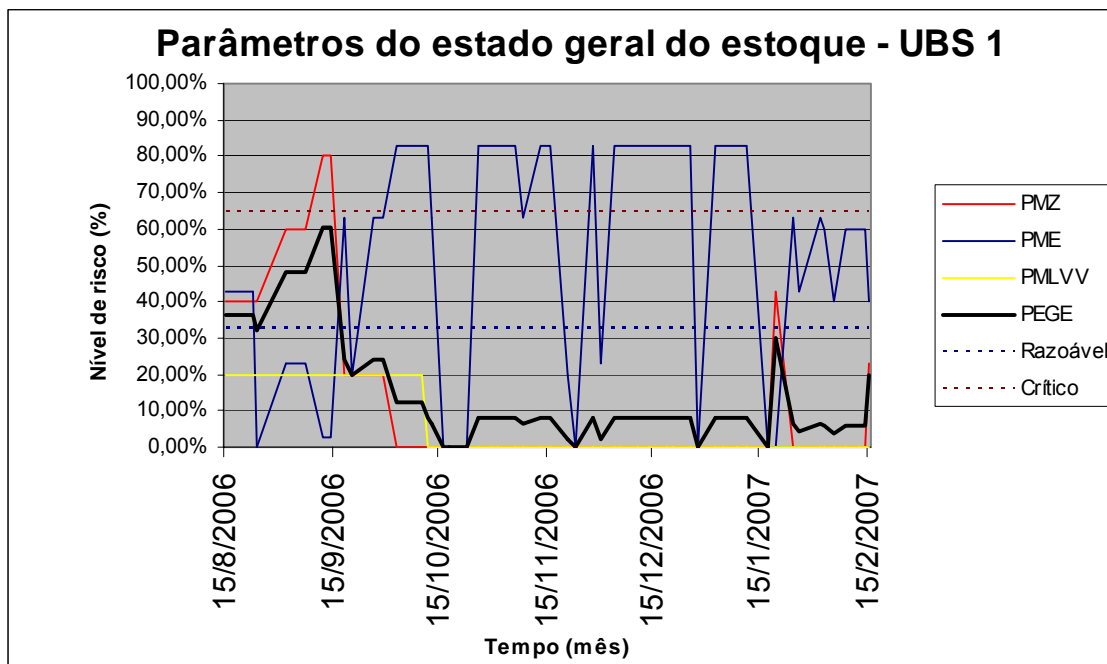


Figura 63 - Estado geral do estoque da UBS 1.

No mesmo período, o percentual de materiais excedentes (PME) conseqüentemente assumiu um valor próximo ao ideal com o nível de risco (NR) tendendo a zero, em alguns momentos. Já para o percentual de materiais com lotes em véspera de vencimento (PMLVV) o NR assumiu um estado estável, em função da considerável minoria de materiais que possuíam lotes em véspera de vencimento. No final do segundo mês, o PMLVV atingiu o estado ideal para o NR (valor zero), em virtude das saídas feitas dos lotes que estavam em véspera de vencimento, através das orientações do AGT.

Para os demais meses, os dois principais parâmetros (PMZ e PMLVV) tenderam ao estado ideal. Entretanto, o percentual de itens excedentes, na maior parte do tempo, atingiu o NR crítico. Na verdade é uma situação menos agravante a ter o NR com valor crítico para o percentual de materiais zerado (PMZ). O NR do PEGE nos meses seguintes oscilou entre faixa de valores razoável e bom, no decorrer de toda a simulação.

Um outro fator importante a ser citado é que em alguns pontos da Figura 63, todos os parâmetros analisados assumiram valor zero, ou seja, o NR com valor ótimo para o estoque. Situação atingida através das devidas doações para os setores requisitantes, bem como a devolução dos materiais que estavam excedentes para o almoxarifado gestor.

Diante de tais fatos, deduz-se que o usuário agiu em concordância com o AGT para tomar as providências, a fim de regularizar o seu estoque.

5.7.3 Análise dos parâmetros do EGE da UBS 2

No início da simulação, o NR dos parâmetros da UBS 2 (representado pela Figura 64) assumiu um valor estável, com exceção do PME, cujo NR possuía um valor crítico, por volta da data de 20/08/2006. Esta transição se deve ao processo de convergência entre o estoque real para o ideal, os quais incluem as operações de doações para os setores requisitantes e devolução destes materiais para o almoxarifado gestor, conforme apresentado nos gráficos: Figura 49, Figura 54 na seção anterior e aos demais gráficos contidos no anexo A para a UBS 2. Ao fim do primeiro ciclo, o PME muda novamente o valor do NR de razoável para crítico, ocasionado pelo processo de entrada de item no setor.

Na segunda metade do segundo mês, o PMLVV atingiu o estado ideal quando o usuário gerou as saídas dos lotes irregulares. Os demais parâmetros mantiveram o mesmo

comportamento que o primeiro mês, com NR razoável. O PEGE, de um modo geral, tendeu ao NR bom.

Entre o quinto e o sexto mês, o PEGE oscilou entre NR de valor bom e razoável, em virtude do PMZ neste período possuir um NR de valor crítico. Embora a menor quantidade de itens da lista de entrada estar zerada (observado na Figura 59), a sua maioria mantinha situação normal, ilustrado pelas figuras: Figura 49, Figura 54 e demais em anexo . No entanto, os itens em estado normal não apresentam peso significativo no cálculo do EGE quando equiparado com os itens irregulares, pois um item zerado possui o maior peso nos cálculos do EGE, ainda mais quando o item irregular é de classificação do tipo Z (em que possui peso máximo no cálculo), sendo um dos fatores que ocasionaram esta discrepância.

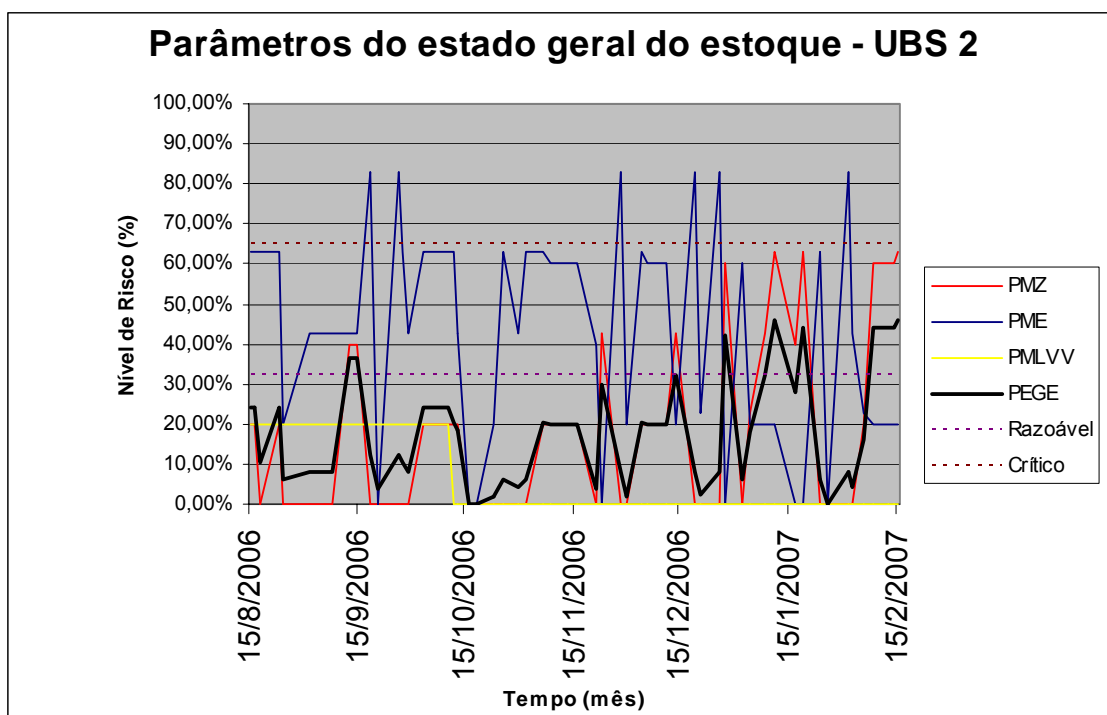


Figura 64 - Estado geral do estoque da UBS 2.

5.7.4 Análise dos parâmetros do EGE da UBS 3

Durante o primeiro mês, a UBS 3 manteve o NR dos parâmetros do EGE com status razoável, tendendo a uma melhora na segunda metade do mês, quando ocorre uma queda do NR do PMZ, tendendo ao status bom, apresentado pela Figura 65. Esta variação do NR decorreu do acerto no estoque, realizado pelo usuário com auxílio do AGT, ilustrado nos gráficos do histórico do estoque da UBS 3, quando é feito o processo de requisições dos materiais zerados, (no caso do materiais 2 e 3, mostrados pelas Figura 55, Figura 60 e materiais 4 e 5 em anexo) e pela doação dos itens excedentes ao almoxarifado, ilustrado pela , Figura 50 do material 1. Note que a UBS 3 não possui itens com lotes em véspera de vencimento, sendo assim, o NR de PMLVV está com valor ideal, tendendo a zero.

No segundo mês, o NR do EGE tende a ótimo, com exceção dos materiais excedentes que atingem o nível crítico.

No terceiro mês da simulação, o NR do PMZ chega a atingir o estado crítico, devido ao fato de a maioria dos itens atingirem a margem dos zerados. Todavia, esta discrepância é sanada quando ocorre o suprimento feito pelo almoxarifado gestor. Para os meses seguintes, o NR dos parâmetros assumem uma certa estabilidade, tendendo ao estado bom.

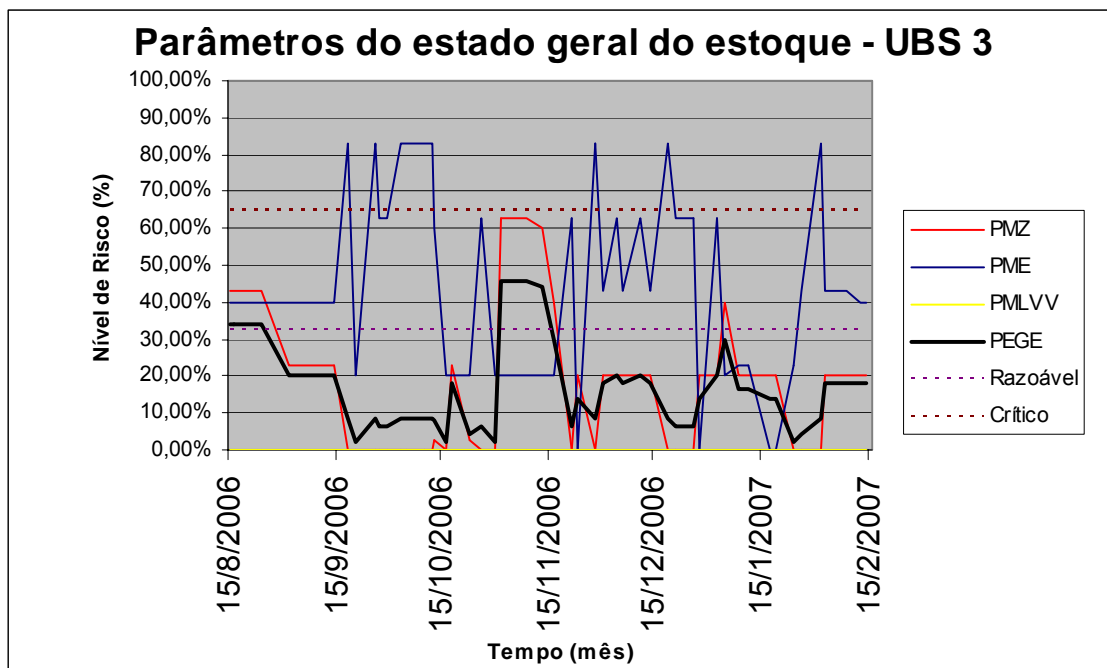


Figura 65 - Estado geral do estoque da UBS 3.

5.7.5 Análise dos parâmetros do EGE da UBS 4

A UBS 4 iniciou o seu primeiro mês de simulação com alguns itens que possuíam lotes em véspera de vencimento e maior parte da lista de suprimento com materiais excedentes, conforme ilustrado pela Figura 66. Entretanto, o PEGE obteve um status de bom em quase todo o período de simulação. Apenas chegou a atingir o status de razoável na véspera da data de suprimento para o segundo mês, em que ocorreu um aumento no NR do PMZ.

No decorrer do primeiro mês, o NR do PME atingiu o estado crítico que, gradativamente, foi convergindo para estado razoável no fim do ciclo. Este fato pode ser observado por volta da data 20/08/2006, em que é notada a redução do NR dos materiais excedentes. Este efeito é causado pela realização das movimentações dos itens em excesso para os setores requisitantes da sociedade de agentes e também pela devolução dos itens ao almoxarifado gestor, apresentado pelas figuras: Figura 51, Figura 56 e gráficos dos materiais 4 e 5 da UBS 4 em anexo. Esta queda pode ser notada no decorrer de todos os ciclos da simulação para todos os setores.

Outro fator importante a ser observado é que no início do primeiro mês havia materiais com lotes em véspera de vencimento e que foram extintos próximo à data de suprimento. O usuário gerou a saída de tais itens de seu estoque, mediante as orientações de alerta do AGT. Durante a simulação, não ocorreu mais indícios de materiais com lotes em véspera de vencimento para o presente setor.

No segundo mês de processamento, o NR do EGE tendeu a subir proporcionalmente, em relação ao PMZ. O PEGE e o PMZ atingiram um status razoável, em virtude do repentino consumo dos itens da lista, se aproximando da situação de nível normal de consumo, mas tendendo à margem dos itens zerados, conforme apresentado no capítulo 4.

Nos meses três, quatro e cinco, a UBS 4 passou a ter uma estabilidade em relação ao estado dos parâmetros, conservando um status final de bom do estoque. Todavia, há a exceção aos materiais excedentes, em que a UBS 4 atingiu um valor crítico em algumas situações.

De um modo geral, no fim de cada ciclo ocorreu um aumento do NR dos materiais zerados. Este fato foi causado pelo aumento do consumo, que chegou a atingir o estoque de segurança (ES) dos materiais.

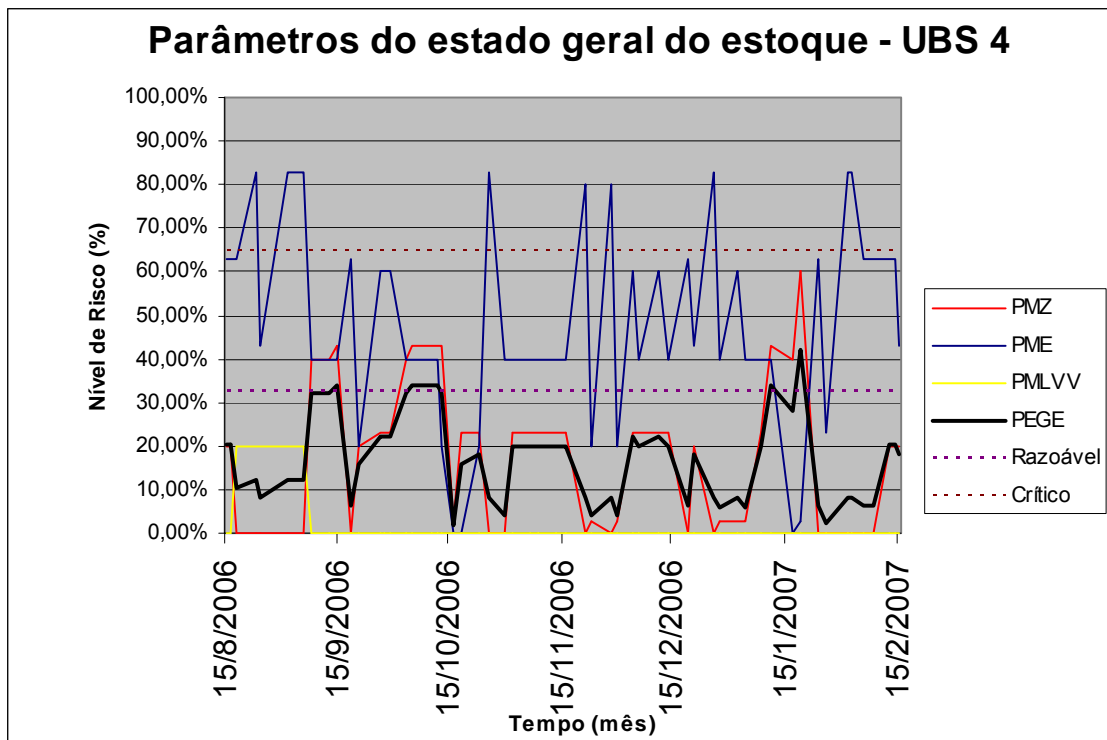


Figura 66 - Estado geral do estoque da UBS 4.

5.7.6 Análise dos parâmetros do EGE da UBS 5

De um modo geral, a UBS 5, representada pela Figura 67, foi o setor da sociedade de AGTs que mais oscilou os parâmetros do estado geral entre os meses de processamento.

No primeiro mês, o NR dos itens excedentes iniciou com um valor crítico, devido à quantidade excessiva contida no estoque, em maior parte dos itens da lista de entrada. Por volta da data do dia 20/08/2006, o NR tendeu a cair, atingindo um status razoável, causado pelo processo de doação de materiais e devoluções para o almoxarifado gestor. No mesmo período, foram tomadas as medidas com relação ao tratamento dos lotes em véspera de vencimento. Note que o indicativo de lotes em véspera de vencimento tendeu a zerar justamente no período em que foram realizadas as saídas dos materiais para os setores requisitantes e almoxarifado gestor, mostrado pelas figuras: Figura 52, Figura 54, Figura 59 e gráficos dos itens 4 e 5 da UBS 5 em anexo.

O PMLVV estava com NR razoável, que passou para o valor bom na segunda metade do primeiro mês. O PMZ manteve o NR zero até a segunda metade do segundo mês, que atingiu um status razoável pela repercussão causada pelas operações de acerto do estoque, realizadas anteriormente.

No terceiro mês o NR do PMZ tendeu a ficar com um status ótimo e na segunda metade atinge o status razoável. Implicitamente indica que nesta transição ocorreu consumo acima do normal, como é o caso apresentado na Figura 52.

Um fato importante a ser notado na UBS 5 é no período da segunda metade do quarto mês da simulação, cuja curva percentual de materiais zerados atingiu um status crítico do NR, causado pela predominância de itens que ficaram abaixo da curva do estoque ideal, conforme Figura 52, Figura 57, Figura 62 e materiais 4 e 5 da UBS 5 em anexo.

O quinto mês chegou a atingir um estado ideal, comparado aos demais. Seus parâmetros atingiram estado ideal próximo à segunda metade do mês, com exceção aos itens excedentes, que manteve um NR crítico, tendendo a uma estabilização do estoque.

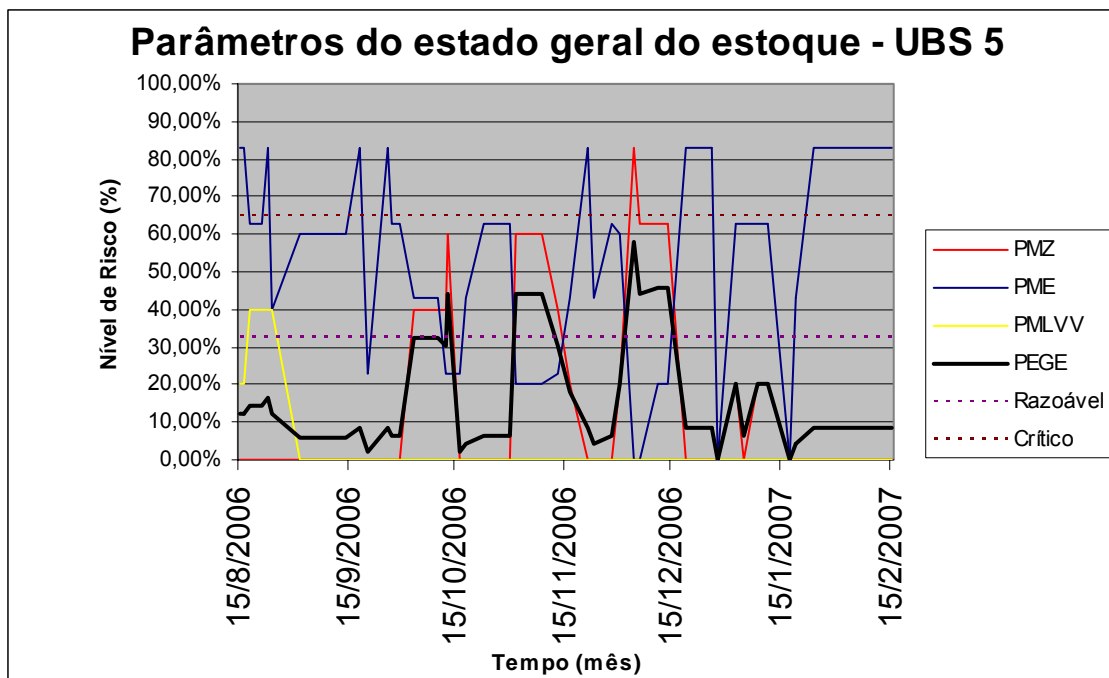


Figura 67 - Estado geral do estoque da UBS 5.

5.8 Considerações finais sobre o EGE perante a sociedade de AGTs

Durante a simulação em todos os setores, a curva do PME convergiu, praticamente, a um valor inversamente proporcional ao PMZ. Esta variação é justificada pelo seguinte fato: quando maioria dos itens da lista de suprimento estão em estado excedente, conseqüentemente a minoria dos itens estão em estado zerado. A situação normal é uma pequena faixa de valores, mediada entre estas duas regiões que definem o estado ideal do estoque. Para os materiais que possuem a faixa de valores fora do estado ideal, o AGT tende a cobrar com maior intensidade o usuário para o seu tratamento, trazendo um melhor padrão de qualidade, com relação aos cuidados do estoque.

De um modo geral, as curvas dos parâmetros do EGE mantiveram um NR bom, com alguns picos no estado crítico, o que justifica que, de um modo geral, os usuários dos setores seguiram as sugestões do AGT, tomando as devidas providências para otimizar o seu estoque.

Conforme pode ser observado nos gráficos apresentados, o PEGE acompanhou a coreografia do PMZ devido a atribuição do maior peso (peso 3) a este. A atribuição dos pesos foi baseada em pesquisas previamente realizadas no estoque distribuído da rede municipal de saúde. Foi feita uma amostragem nos estoques dos setores para identificar e classificar os principais parâmetros que causam um maior impacto no estoque, ou seja, parâmetros que são os principais focos de problemas encontrados da rede municipal de saúde. A partir desta pesquisa, foi nomeado o PMZ como sendo o parâmetro mais importante a ser tratado no estoque e, conseqüentemente, os parâmetros PMLVV e PME.

No início da simulação, as oscilações do NR do EGE e do estoque real eram freqüentes, em virtude do gradativo processo de convergência do estoque, que ganhou uma certa estabilidade ao fim das simulações, obtendo como produto final um NR tendendo ao estado bom e um estoque mais coerente às realidades da rede municipal.

Com o cálculo do EGE tornou-se possível realizar uma análise do estoque sob vários aspectos. Possibilita-se inferir sobre as ações realizadas pelo usuário, obtendo uma conclusão concreta sobre sua atuação no seu respectivo setor.

6 Discussão e Conclusões

6.1 Contribuições

O uso da sociedade de Agentes Gestores Tutores (AGTs), modelados sobre a arquitetura de Sistemas Tutores Inteligentes (STI), trouxeram uma série de contribuições, tais quais:

Ferramenta genérica de apoio à decisão para sistemas de e-Gov (no caso, gestão de materiais)

Este trabalho gerou como contribuição uma ferramenta genérica de apoio à decisão, de caráter educativo, para os sistemas de e-Gov, com a finalidade de monitorar e orientar usuários em processos que exigem o emprego de elementos da área de administração. Como estudo de caso, foi utilizado o protótipo como suporte à gestão de materiais de uma rede municipal de saúde.

Aplicação da tecnologia de STI sob uma nova vertente

O uso de conceitos provenientes da tecnologia de STI, modelada a trabalhar sobre a administração de materiais, mais especificamente na distribuição e gestão de materiais na rede municipal de saúde, levantou novos conceitos sobre a aplicação deste tipo de arquitetura. A arquitetura implementada demonstrou como sendo viável em relação aos resultados esperados. As estratégias pedagógicas do AGT auxiliaram de maneira eficiente, gerando alguma espécie de conhecimento para o usuário sobre o estoque de materiais e princípios de sua gestão, ora visto como uma forma de aprendizado, pois aprender consiste no “ato de adquirir conhecimento ou habilidade prática sobre algo”, [HOU1999].

Proximidade com o comportamento do ser humano

Gerar um AGT embasado em STI possibilitou produzir uma solução próxima a um humano profissional agente gestor, devido ao fato dos STIs possuírem uma característica peculiar, que é adaptar-se ao usuário através das ações que são realizadas por este e são armazenadas no modelo do aprendiz. As ações são convertidas em variáveis em que o tutor possa entender e inferir sobre elas definindo o perfil de gestor deste usuário, desta forma, o tutor obtém subsídios para auxiliar o usuário nas tarefas de gerir um estoque de forma eficaz, identificando as suas deficiências e oferecendo sugestões de tratamento do estoque.

A arquitetura do AGT, ora projetada sobre a estrutura de agentes dotados de estados mentais, como a arquitetura BDI (arquitetura de crenças, desejos e intenções), possibilitou ao AGT um sistema de planejamento baseado em conceitos da psicologia cognitiva, uma “forma de raciocínio” mais próxima do comportamento humano para realizar tarefas inteligentes, como a gestão de materiais.

Estudo sobre a linguagem de programação orientada a agentes e padrões de comunicação

Este trabalho pode ser visto como uma revisão sobre a arquitetura de agentes e também como uma referência sobre a programação de agentes cognitivos. Estes últimos, modelados sob a arquitetura de estados mentais, através da linguagem de programação *AgentSpeak(L)*, coforme [RAO1992] [WEI1999] [WOO2000], e também sob o protocolo de comunicação entre agentes KQML,[FIN1992] [FIN1994a] [GEN1994].

Com o protótipo, foi possível demonstrar, de forma satisfatória, a viabilidade do uso da combinação das ferramentas Jason e SACI na programação de agentes cognitivos e implementação de uma organização social de agentes que se comunicam entre si para realizar um trabalho cooperativo na resolução de um problema complexo e de natureza distribuída, uma vez que as duas ferramentas são integráveis, conforme Hübner [HUB2003] e [HUB2004]. Além do mais, a ferramenta Jason possibilitou desenvolver parte da arquitetura do agente (funções de seleção de planos, percepção, revisão de crenças, comunicação e

atuação) na linguagem Java. Por estes fatores, tornou-se possível desenvolver agentes BDI organizados em três camadas bem definidas, tais quais: camada cognitiva; de sensoriamento e atuação; e comunicação.

Outra característica importante é, que a ferramenta Jason possui uma interface de desenvolvimento das estruturas de planos em *AgentSpeak(L)* dos Agentes BDI, o que facilitou consideravelmente na construção e depuração dos planos e execução das ações dos agentes.

Contribuições sociais

Com base nos resultados obtidos, fruto da simulação de seis meses de processamentos, podemos observar que o trabalho cooperativo da sociedade de STI auxiliou os usuários das UBSs a resolverem a questão da carência e o excesso de medicamentos nos setores, trazendo um certo equilíbrio ao estoque geral e seu consumo. Dessa forma, cria-se uma série de contribuições para a rede municipal de saúde, tais quais:

- A prevenção de gastos em demasia com medicamentos;
- A alta rotatividade de medicamentos, o que evita que os itens pereçam nos setores, gerando um maior giro no capital investido;
- Possibilidade de redução do estoque do almoxarifado centralizador, situação ideal para um estoque, conforme Viana [VIN2006];
- Um controle mais completo do estoque de modo geral.

Outro ponto relevante é que foi comprovado que a carência de medicamentos nos setores não é por falta no estoque geral, mas sim, pelas falhas na distribuição dos medicamentos, ocasionadas pelas solicitações equivocadas de cotas.

Diante destes fatos, pode-se dizer que é uma proposta atrativa o uso de STI, modelados sob sistemas Multi-Agentes cognitivos, como ferramenta auxiliar para o monitoramento e gestão de estoque, atuando na resolução da distribuição de medicamentos, a qual ainda é um dos principais problemas encontrados nas redes municipais de saúde.

6.2 Trabalhos Futuros

Proposta de Arquitetura Agentes Facilitadores Móveis

Uma das limitações encontradas na arquitetura implementada é sobre o uso de agentes facilitadores para gerir a comunicação entre os agentes das sociedades.

A arquitetura de comunicação orientada a agentes facilitadores, apesar de ser uma das alternativas mais bem aceitas pela comunidade de agentes para a construção de Sistemas Multi-Agentes (SMA), possui uma certa limitação. Quando uma estação de agente facilitador cai, nenhum agente da sociedade consegue se comunicar, congestionando todo o sistema. Desta forma, cabe, como um trabalho seqüente, produzir uma arquitetura estendida de agentes facilitadores que trate tal limitação.

Se o agente facilitador da sociedade cair, é possível gerar novos agentes facilitadores que tratem tal fato. Quando um agente entra na sociedade, este recebe do agente facilitador uma lista que contém todos os endereços lógicos dos agentes da sociedade, ordenados por entrada. Através desta lista, é possível levantar um novo agente facilitador, a partir do primeiro agente que entrou na sociedade, disparando um processo de criação do agente facilitador.

Modificações no cálculo do Estado Geral do Estoque

Atualmente, o cálculo do estado geral do estoque está sendo feito apenas pela classificação de materiais XYZ. Este cálculo poderia ser ampliado para a classificação ABC de materiais, tipo de classificação orientado ao valor do material em relação ao estoque, combinadas com a classificação XYZ. Desta forma, os materiais que possuem a classificação A e Z seriam os de maior valor e peso no estoque, conseqüentemente, necessitariam de maior atenção.

Uma outra proposta seria prover a ampliação da lista de parâmetros que compõem o Estado Geral do Estoque (EGE), cujos parâmetros tratados atualmente são apenas três: os materiais excedentes, os zerados e os que possuem lotes em véspera de vencimento.

Com o aumento da quantidade de parâmetros, o modelo do aprendiz se torna mais completo, acarretando uma quantidade maior de dados, em que o AGT possa realizar uma análise mais concreta sobre o perfil do aprendiz, tendo uma melhora significativa na qualidade de orientação. Uma solução consiste em abordar parâmetros que envolvam os materiais que estão sendo pouco utilizados no estoque, que possivelmente irão vencer ou permanecer estocados.

A modelagem matemática do EGE foi implementada usando um sistema de médias ponderadas, de forma que os pesos para cada variável foram previamente definidos e fixados. Como uma proposta futura, este modelo poderia ser substituído por um modelo dinâmico, como por exemplo, o uso de uma rede neural em que os pesos poderiam ser ajustados de forma dinâmica, trazendo resultados mais interessantes para o EGE.

Atualmente, o AGT foi construído para trabalhar com a camada de comunicação e cognitiva juntas, o que afeta o seu desempenho. Tais camadas poderiam ser separadas em dois agentes: um agente cognitivo, responsável por realizar todo o processo de tomada de decisão; e um agente comunicador, que tem a função de prover a comunicação entre os demais agentes, aguardando por uma mensagem e enviando mensagens do agente facilitador.

6.3 Bibliografia

[FIN1992] Finin, T., Fritzson, R., and McKay, D. *A language and protocol to support intelligent agent interoperability*. In Proceedings of the CE & CALS Washington Conference.1992.

[FIN1994a] Finin, T., Fritzson, R., McKay, D., McEntire, R. *KQML as an agent communication language*. In Proceedings of the Third International Conference on Information and Knowledge Management. ACM Press, 1994a.

[GEN1994] Genesereth, M. R. and Ketchpel, S. P. *Software agents. communications of the ACM*, 1994. 37(7):48–53.

[HOU1999] Houaiss, A.; Villar, M. S. *Minidicionário Houaiss da Língua Portuguesa*. 2º Ed., Rio de Janeiro – RJ, 1999. Editora: Objetiva.

[HUB2003] Hübner, J. F., Sichman, J. S. *Saci Programming Guide*, USP - Universidade de São Paulo, São Paulo, jul 2003.

[HUB2004] Hübner, J. F., Bordini, R. H., Vieira, R. *Introdução ao desenvolvimento de Sistemas Multiagentes com Jason*. Departamento de Sistemas e Computação, Universidade Regional de Blumenau (FURB). 2004, p. 1-12.

[RAO1992] Rao, A. S. e Georgeff, M. P. *An abstract architecture for rational agents*. In Rich, C., Swartout, W. R., e Nebel, B., editors, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, 25–29 October, Cambridge, MA, 1992. pages 439–449, San Mateo, CA. Morgan Kaufmann.

[VIN2006] Viana, J. J. *Administração de Materiais: Um Enfoque Prático*. 1º Ed., São Paulo – SP, 2006. Editora: Atlas.

[WEI1999] Weiss, G. *Multyagent Systems: A Modern Approach to Distributed Artificial Intelligence*, 1999, p. 121 Ed. Gerhard Weiss.

[WOO2000] Wooldridge, M. J. *Reasoning about Rational Agents*. The MIT Press, Cambridge, MA, 2000.

Anexo A – Histórico do Estoque dos Materiais das UBSs

A.1. Histórico do Estoque dos Materiais 4 e 5 da UBS 1

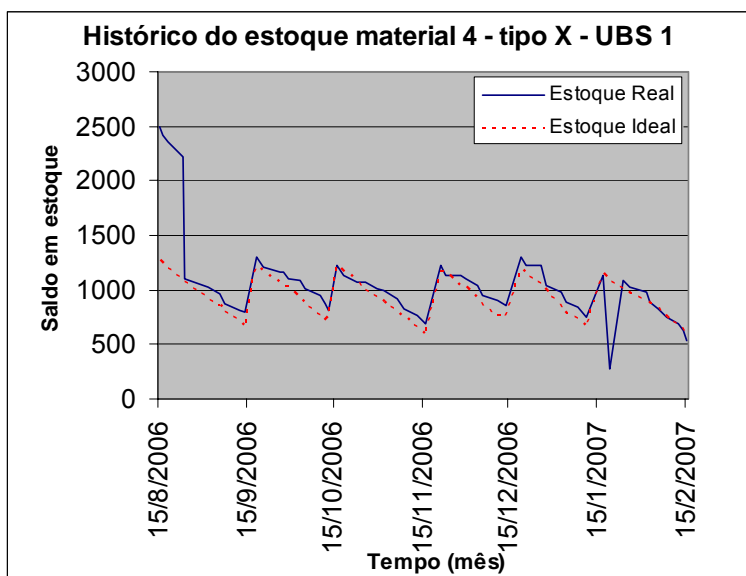


Figura 68 - histórico do estoque do material 4, UBS 1.

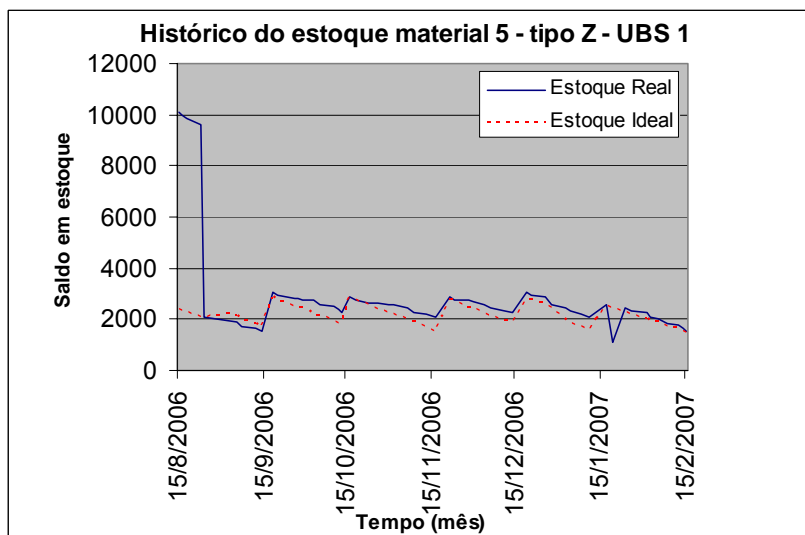


Figura 69 - histórico do estoque do material 5, UBS 1.

A.2. Histórico do Estoque dos Materiais 4 e 5 da UBS 2

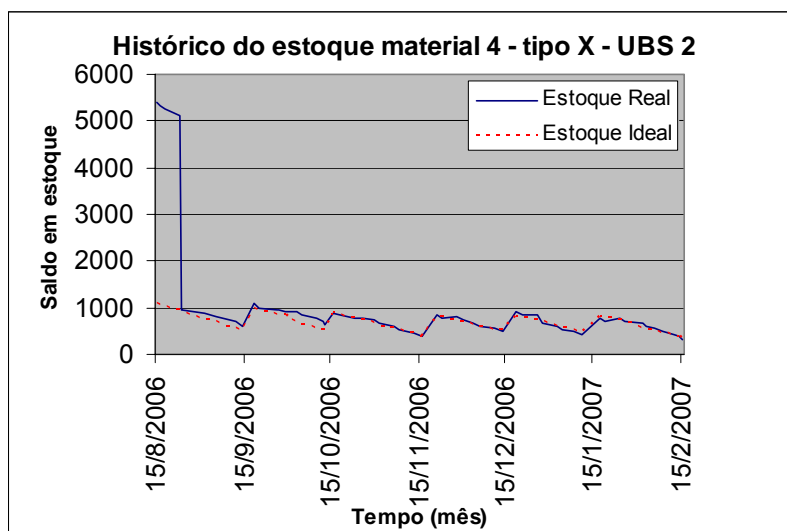


Figura 70 - histórico do estoque do material 4, UBS 2.

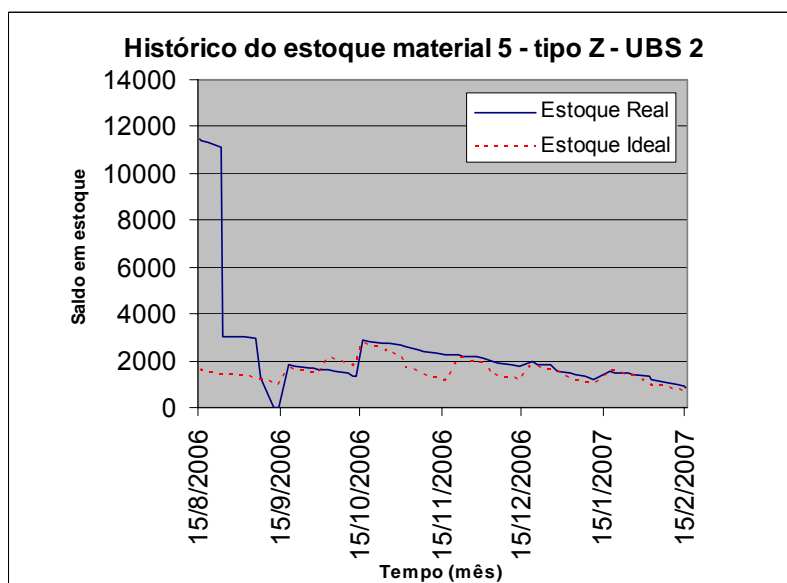


Figura 71 - histórico do estoque do material 5, UBS 2.

A.3. Histórico do Estoque dos Materiais 4 e 5 da UBS 3

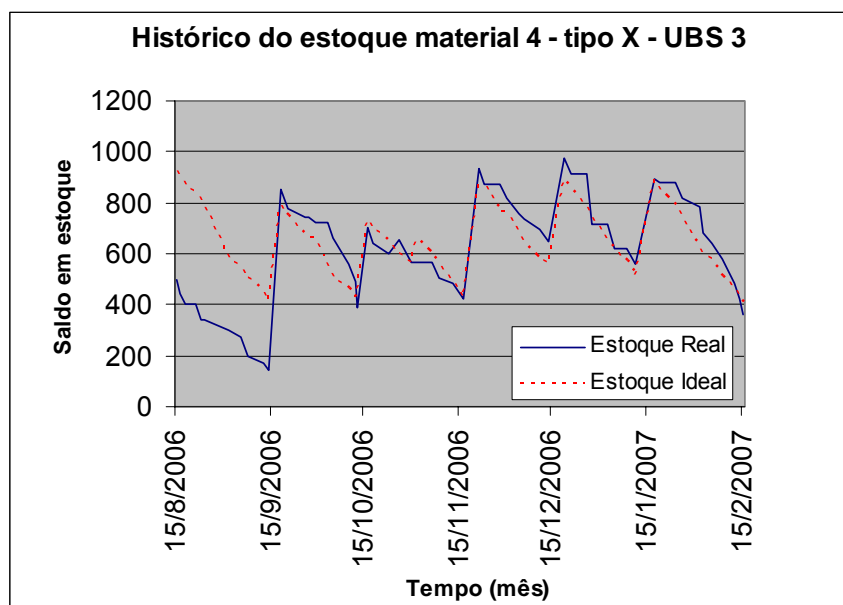


Figura 72 - histórico do estoque do material 4, UBS 3.

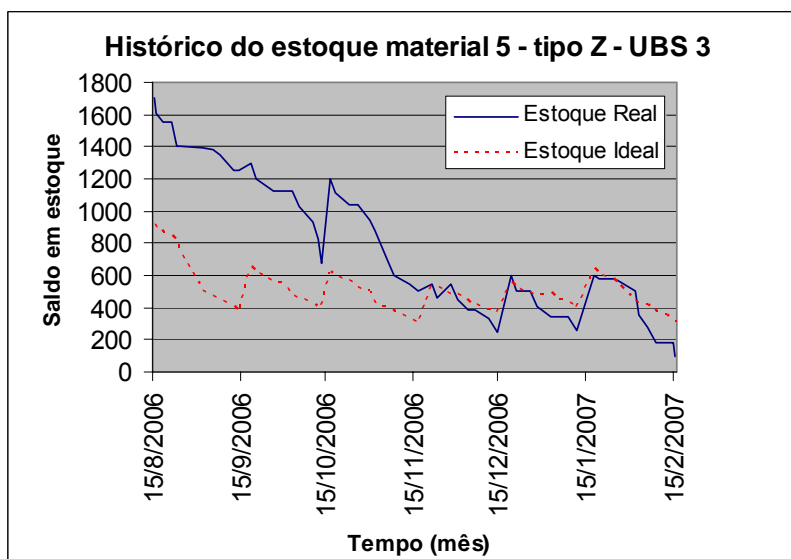


Figura 73 - histórico do estoque do material 5, UBS 3.

A.4. Histórico do estoque dos materiais 4 e 5 da UBS 4

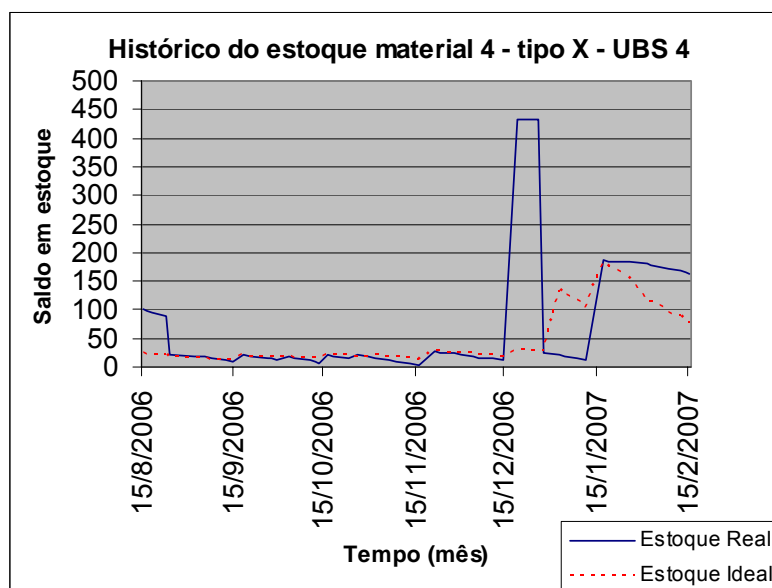


Figura 74 - histórico do estoque do material 4, UBS 4

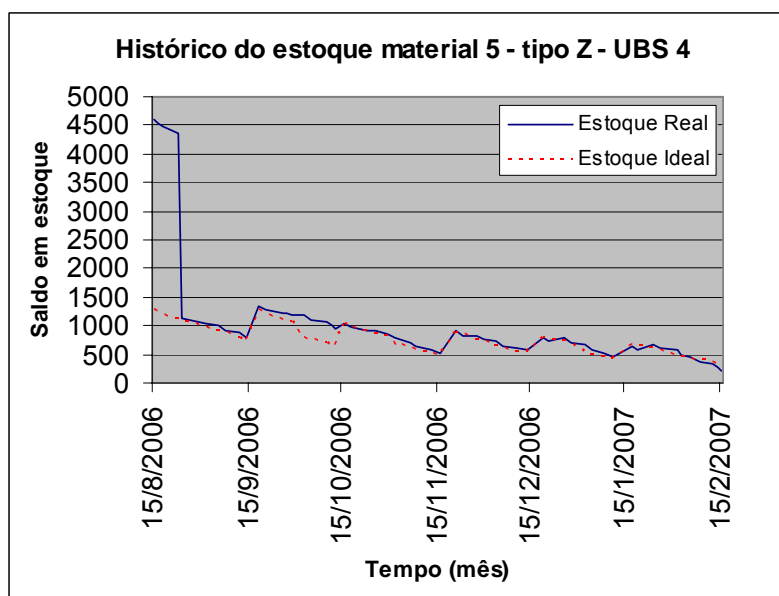


Figura 75 - histórico do estoque do material 5, UBS 4

A.5. Histórico do estoque dos materiais 4 e 5 da UBS 5

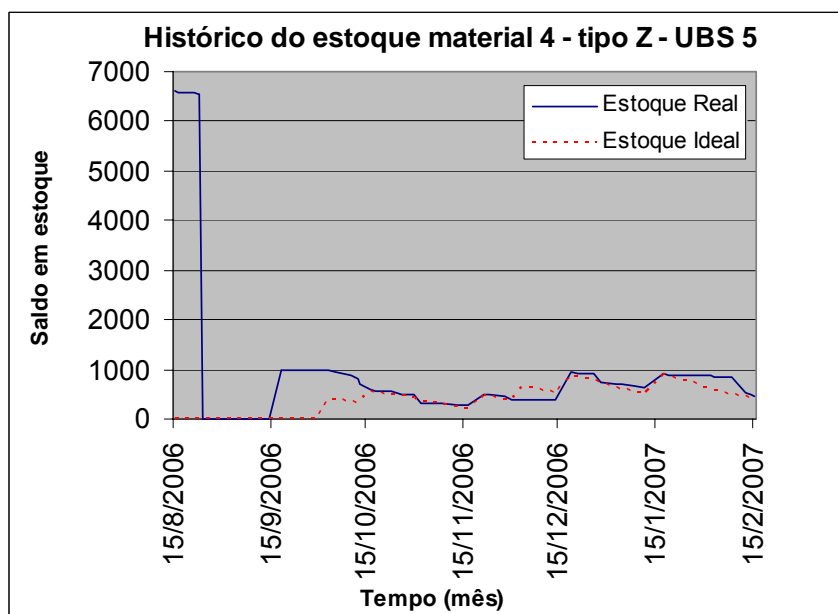


Figura 76 - histórico do estoque do material 4, UBS 5

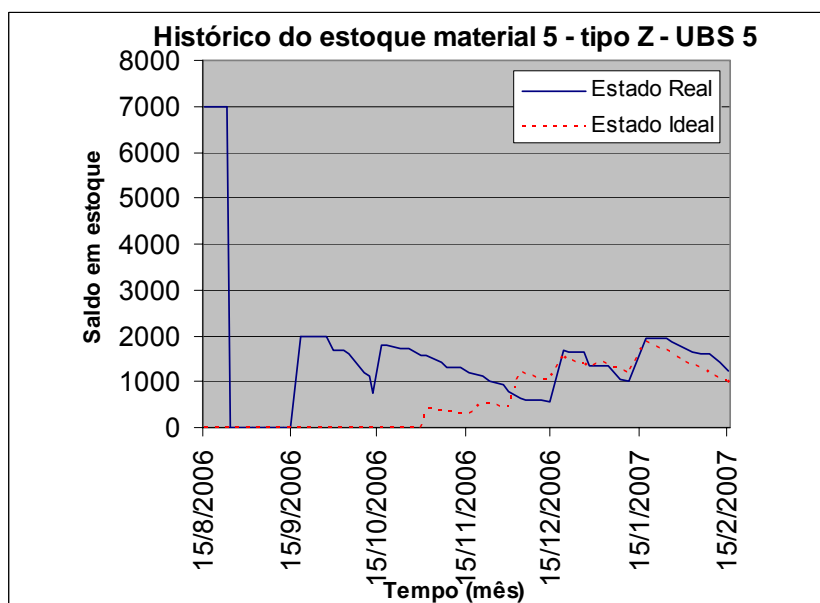


Figura 77 - histórico do estoque do material 5, UBS 5

Anexo B - Gestão de materiais e medicamentos

B.1. Introdução

Este capítulo aborda alguns conceitos de gestão de materiais, que são comumente utilizados, bem como procedimentos de gestão, a fim de serem aplicados sob a forma de ações na arquitetura de STI para o tratamento dos estoques dos subalmoxarifados da rede municipal de saúde.

B.2. Gestão de Estoque Função e Atividades

B.2.1. Fundamentos

Gestão é o conjunto de atividades e ações que têm por função gerar o pleno atendimento das necessidades da empresa com o máximo de eficiência possível e menor custo, através do maior giro possível para o capital investido em materiais. A área de gestão possui técnicas de tratamento que objetivam buscar um equilíbrio entre o estoque e o consumo, intermediado por variáveis de ressuprimentos, que acompanham dinamicamente a evolução do estoque. Com os procedimentos de gestão de estoque, torna-se possível estimar quanto adquirir de materiais e quando, para repor o estoque gradativamente, repercutindo em um estoque próximo do ideal.

Segundo Viana [VIN2006], o equilíbrio entre estoque e consumo pode ser obtido através de alguns critérios. São estes:

- Impedimento de entrada de materiais desnecessários, conservando em estoque apenas os materiais de real necessidade;
- Centralização das informações que possibilitem o pleno acompanhamento e planejamento das atividades de gestão;

- Desenvolver e implantar a política de padronização de materiais, bem como de sua nomenclatura;
- Realizar a definição dos parâmetros de administração de materiais necessários para cada material, como: definição do estoque máximo e de segurança, ponto de pedido e quantidade a ser comprada. Esta tarefa pode ser realizada através de cálculos relativos à gestão de materiais;
- Fazer o acompanhamento e análise da evolução dos estoques da empresa, desenvolvendo estudos estatísticos;
- Monitorar periodicamente o estoque, na tentativa de buscar materiais que sejam obsoletos para em seguida retirá-los.

Para um melhor entendimento sobre o processo de gestão de materiais, é necessário conhecer, pelo menos basicamente, o estudo da natureza e classificação dos materiais, bem como os componentes que regem os processos de compra, recebimento, entrada, saída e inventário.

Na próxima seção, apresentaremos uma breve descrição sobre estes elementos. Antes, porém, precisamos definir o conceito de “Almoxarifado”.

B.2.2. Almoxarifado

Almoxarifado é a entidade responsável por realizar a guarda dos materiais, garantindo a sua preservação e integridade até o consumo final, a distribuição. De acordo com Araújo [ARA1975], a palavra *almoxarifado* passou a ser usada como “referência a tudo que se refira à ordem, disciplina, registros de fatos, previsões e provisões, não somente de materiais e matérias-primas, como tudo o que diz respeito à perfeita ordenação, da forma pela qual decorrem os fatos e as previsões como deverão ser”.

O almoxarifado deve ser flexível o bastante para acompanhar as condições de armazenagem e organização e ter a capacidade de armazenar materiais, de acordo com a sua demanda. Por exemplo, para o armazenamento de gases é necessário que se tenha recipientes adequados resistentes à pressão. Já materiais sensíveis à temperatura devem ser armazenados em câmaras adequadas.

B.3. Componentes da administração de materiais

Para um melhor entendimento sobre os processos de gestão de materiais, é relevante que antes se conheça algumas definições importantes, intimamente relacionadas à gestão como conceitos de estoque, consumo e demanda de materiais.

B.3.1. Estoque

O significado do substantivo *estoque* é bastante abrangente. De acordo com Viana [VIN2006], *estoque* consiste na representação explícita de matérias-primas, produtos semi-acabados, componentes para montagem, sobressalentes, produtos acabados, materiais administrativos e suprimentos de uma forma variada, etc. Já Martins [MAR2006] aborda o conceito de *estoque* como elemento regulador do fluxo de produção no ramo manufatureiro, ou do fluxo de vendas, quando se trata do processo comercial.

Perante tais definições, pode-se dizer que estoque é tudo que se refere a materiais, mercadorias ou produtos acumulados, para serem utilizados à medida que são requisitados. Isto permite o atendimento de acordo com as necessidades do usuário, ou seja, uma reserva para ser utilizada quando necessário.

B.3.2. Consumo

O consumo é uma representação da quantidade de materiais requerida para o atendimento das necessidades de produção ou comercialização em uma determinada unidade de tempo. São três formas de consumo:

- Consumo regular – tipo de consumo caracterizado pela pequena variação de sua quantidade em uma série temporal, levando em consideração o tempo em intervalos constantes;
- Consumo irregular – tipo de consumo caracterizado por materiais utilizados em quantidades aleatórias, em uma série temporal;

- Consumo sazonal – tipo de consumo caracterizado pelo padrão repetitivo da demanda, mas que, em alguns períodos, ocorre considerável elevação no consumo.

B.3.3. Demanda

A demanda é uma medida de expectativa de consumo dos materiais, levando em consideração a sua evolução histórica. Seu conhecimento permite prever as estimativas futuras de materiais que serão consumidos ou comercializados por uma empresa.

Quando ocorre a entrada de um material pela primeira vez no estoque, a demanda é estimada pelo próprio usuário, ocasionando distorções devido à falta de dados sobre o consumo. Após a ocorrência de movimentações de estoque, a demanda passa a ser calculada de forma dinâmica, a partir de algumas variáveis, como o histórico de consumo, tempo de reposição e critérios de classificação de um material, tornando possível realizar previsões sobre as demandas.

B.3.4. Previsões de demanda

São conhecidas também como previsões de consumo. De acordo com Bowersox [BOW1999], são projeções de valores e quantidades que provavelmente serão produzidas, vendidas, expedidas e ou consumidas. O levantamento de previsões precisas permite equilibrar as demandas por recursos e minimizar seus picos, tanto de capacidade quanto de estoque.

Dias [DIA1985] relata que a previsão é o ponto de partida de todo o planejamento empresarial. É necessário compreender que a previsão não consiste em uma meta de vendas, mas deve ser compatível com o custo de obtê-la, ou seja, para que seja satisfatório é necessária a fiel alimentação dos dados por parte do usuário.

As técnicas de previsão da demanda podem ser classificadas em três grupos distintos:

- **Projeção:** técnica de natureza quantitativa, que se baseia no passado para construir o futuro. Este grupo de técnicas define que as tendências de vendas ou consumo do estoque futuro seguirão explicitamente as ações acontecidas no passado;
- **Explicação:** técnicas que utilizam métodos de regressão e correlação. Procura-se explicar o consumo ou vendas realizadas no passado, relacionando com outras variáveis em que a sua evolução é conhecida, ou ao menos prevista;
- **Predição:** técnica em que se utiliza profissionais experientes no ramo da logística, conhecedores dos fatores implícitos e explícitos que influenciam nas vendas ou nos consumos e, através de tais fatores, realizam sua previsão de demanda.

Existem várias técnicas que contemplam a projeção da demanda, tais como: Método do último período, Método da média ponderável, Método da média móvel, Método da média móvel ponderada, Método da média com ponderação exponencial, Método dos mínimos quadrados. No entanto, neste trabalho iremos tratar apenas do *Método dos mínimos quadrados*, por ser considerado o método mais usando entre as técnicas de projeção.

B.3.4.1. Método dos Mínimos Quadrados

O Método dos Mínimos Quadrados tem por finalidade determinar a melhor linha de ajustamento, uma função $f(x)$ que passe o mais próximo possível de todos os dados amostrados [DIA1985] [DOR1981] [SPI1977].

Para uma melhor compreensão, observe o seguinte exemplo, na execução experimento, em que foi obtida uma tabela de valores x e y . Ao observar o gráfico na Figura 78, podemos observar que existem dois pontos (com asterisco) que possuem um erro em consideração ao eixo y , o qual não segue o padrão dos demais, um erro estatístico. Para que possamos utilizar tais dados, é necessário fazer um ajustamento deste conjunto, de forma a atenuar estes erros.

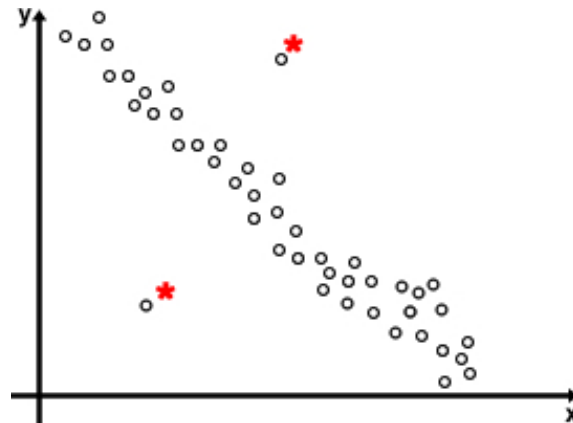


Figura 78 - Exemplo de amostragem de dados agrupados cujo um relacionamento funcional ($f(x)$) aproximado é procurado.

Considerando que uma função $\bar{y} = f(x)$ faz o relacionamento entre x e y , onde \bar{y} indica que é um valor aproximado de y , a atenuação dos erros estatísticos é feita quando se é encontrada uma função que minimize o erro (conhecido também como desvio). O desvio é a diferença entre o valor verdadeiro e o valor aproximado, isto é, $d = y - \bar{y}$, ou seja, para o conjunto de pontos apresentados na Figura 78, existem os seus respectivos desvios, apresentados pela Figura 79.

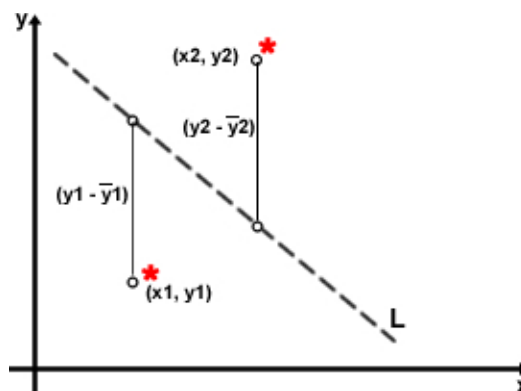


Figura 79 - Apresentação dos desvios do exemplo dos dados amostrados.

Sendo assim:

$$d_1 = y_1 - \overline{y_1} = y_1 - f(x_1) \quad (10)$$

$$d_2 = y_2 - \overline{y_2} = y_2 - f(x_2)$$

$$d_3 = y_3 - \overline{y_3} = y_3 - f(x_3)$$

M

$$d_m = y_m - \overline{y_m} = y_m - f(x_m)$$

Entretanto, não é conhecida a função f que melhor ajuste a cada ponto, ou seja, uma função em que os seus desvios tendem a zero, e a soma dos desvios (D) seja mínima, isto é:

$$D = \sum_{i=1}^m d_i = \sum_{i=1}^m (y_i - \overline{y_i}) \quad (11)$$

Esta definição pode ser melhor visualizada ao observar a Figura 79, em que tem-se dois pontos ($m = 2$) com coordenadas (x_1, y_1) e (x_2, y_2) . Se considerar a função $f(x)$ como uma reta,

$$\overline{y} = a + bx \quad (12)$$

naturalmente, a “melhor” reta a ser selecionada é aquela que passa através de ambos os pontos, porque se calcularmos o (11) para esta reta, dará zero. No entanto, a soma dos desvios d_1 e d_2 assume um valor zero, devido ao problema dos sinais dos desvios serem opostos, o cálculo do $d_1 = y_1 - \overline{y_1}$ resulta em um valor negativo e $d_2 = y_2 - \overline{y_2}$ é positivo, se considerarmos que ambos possuem a mesma grandeza, ou seja, a mesma distância da reta L.

Para superar o problema dos sinais opostos dos desvios, deve-se acrescentar o quadrado na soma dos desvios em (11), o que resulta na *equação da soma dos mínimos quadrados*.

$$D_{\text{quadrado}} = \sum_{i=1}^m d_i^2 = \sum_{i=1}^m (y_i - \overline{y_i})^2 \quad (13)$$

Substituindo o $\overline{y_i}$ em (13) pela equação da reta, a equação passa a ter dois parâmetros a e b , resultando na seguinte equação:

$$D_{\text{quadrado}} = \sum_{i=1}^m d_i^2 = \sum_{i=1}^m (y_i - a - bx_i)^2$$

Para minimizar a equação dos mínimos quadrados é aplicado métodos de cálculos de derivação parcial, em que são determinados os parâmetros de a e b , conhecidos como equações normais da reta de mínimos quadrados definidos em:

$$a = \frac{\sum y_i \cdot \sum x_i^2 - \sum x_i \cdot \sum x_i y_i}{m \sum x_i^2 - (\sum x_i)^2} \quad (14)$$

$$b = \frac{m \sum x_i y_i - \sum x_i \cdot \sum y_i}{m \sum x_i^2 - (\sum x_i)^2} \quad (15)$$

Regressão Linear

A *Regressão Linear* é o método estatístico utilizado para realizar previsões sobre uma variável y , correspondente ao conhecido de uma variável x [DOR1981]. Isto pode ser alcançado mediante a avaliação do valor de y , a partir de uma curva de mínimo quadrado, que se ajusta aos dados amostrais. A curva resultante é denominada de regressão de y para x , visto que y é avaliado a partir de x .

Para um melhor entendimento, observe o seguinte exemplo, (extraído de [SPI1977]). A Figura 80 apresenta a produção de aço nos E.U.A, em milhões de toneladas. Se quisermos realizar a previsão da produção de aço para o ano de 1957, basta aplicar a equação da reta (12) e encontrar os parâmetros a e b através das equações normais da reta. A equação da reta minimizada fica: $y = 75,2 + 3,95x$, que tem $x = 0$ como origem a data de 1946, e $x = 11$ para a data de 1957. Aplicando os valores na equação da reta, é encontrada a previsão de produção de aço para 1957 na grandeza de $y = 118,65$ milhões de toneladas.

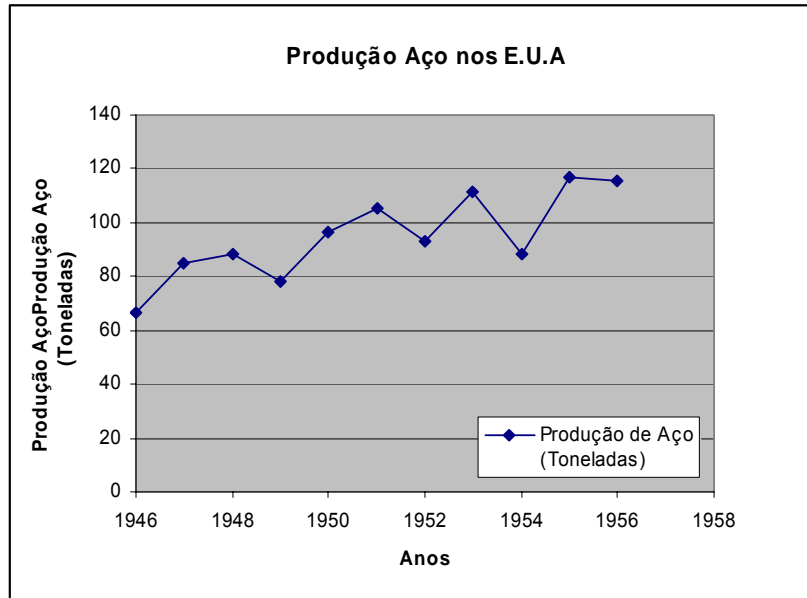


Figura 80 -Gráfico da Produção de aço nos EUA.

B.4. Classificação de materiais

A classificação é o processo pelo qual se agrupam materiais que possuem características semelhantes. Sob o ponto de vista de gestão do estoque, a classificação de materiais é o ponto crucial para o sucesso ou insucesso da gestão. O fato de se poder definir o grau de importância operacional com relação a um grupo possibilita a obtenção de informações relevantes para realizar futuras aquisições e efetivas saídas de materiais desnecessárias do estoque.

B.4.1. Tipos de classificação

Basicamente, existem vários tipos de classificação de materiais que, analisados em conjunto, auxiliam nos processos de tomada de decisão sobre os resultados e contribuem para atenuar o risco de falta. Neste contexto será apresentado apenas um dos principais métodos de classificação, conhecido como classificação por tipo de demanda.

Classificação por tipo de demanda

Uma das principais classificações atribuídas a materiais é a classificação por tipo de demanda, conforme ilustrado pela Figura 81. Na classificação por tipo de demanda, os materiais são divididos em materiais de estoque e de não estoque.

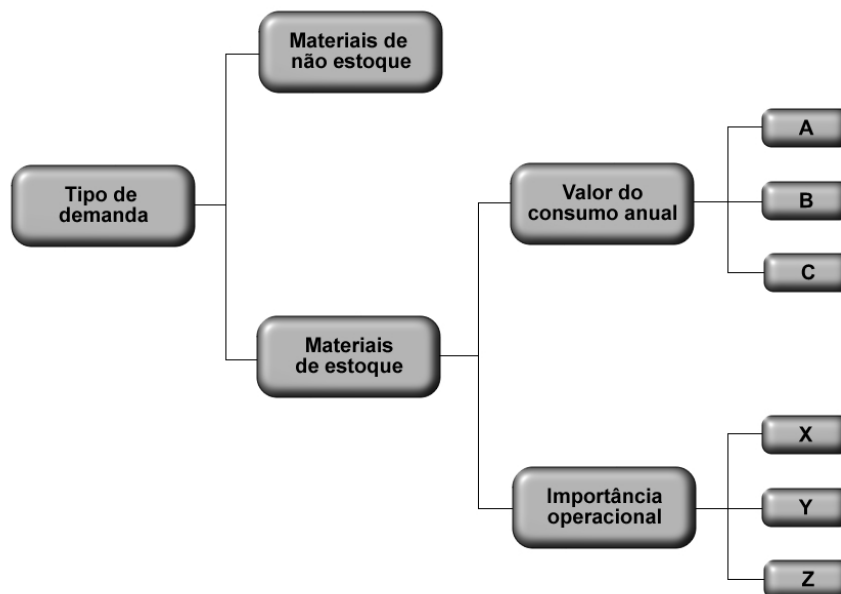


Figura 81 - Classificação por tipo de demanda, extraído de [VIN2006].

Materiais de não estoque

São materiais de demanda imprevisível, para os quais não são definidos parâmetros para o ressuprimento automático. Em virtude de sua irregularidade de consumo, a aquisição destes materiais é feita somente em situações cujo usuário constate a sua necessidade. Estes materiais possuem a característica de serem comprados e consumidos de forma imediata.

Materiais de estoque

Consistem em materiais que devem existir no estoque e aos quais são atribuídos critérios de ressurgimento automático, mediante o conhecimento de uma previsão de demanda.

Os materiais de estoque estão classificados em três subgrupos:

1. Quanto à aplicação:

- a. Materiais produtivos: materiais ligados direta ou indiretamente ao processo de fabricação, tais como matérias-primas, produtos em fabricação, etc;
- b. Matérias-primas: materiais básicos que fazem parte do processo produtivo da empresa;
- c. Produtos em fabricação: materiais que estão sendo produzidos ao longo do processo da empresa;
- d. Produtos acabados: são os produtos prontos, constituintes do estágio final de produção;
- e. Materiais de manutenção: materiais de consumo utilizados na manutenção nos setores da empresa;
- f. Materiais de consumo geral: materiais de consumo geral em diversos setores da empresa.

2. Quanto ao seu consumo anual: refere-se à classe de materiais voltados a sua valorização, em função do seu consumo anual conhecida como *classificação ABC*, conforme Costa [COS2002] e Viana [VIN2006]. Estes materiais podem ser divididos em:

- a. Materiais A: compreendem materiais de grande valor de consumo. Detêm valores em torno de 70% do valor anual de consumo. Requer uma atenção especial pelo administrador;
- b. Materiais B: abrangem materiais de médio valor de consumo. Detêm cerca de 20 % do valor anual de consumo. Intermediário entre os materiais A e C.
- c. Materiais C: são materiais de pequeno valor de consumo. Atinge, no máximo, cerca de 10 % do valor anual de consumo.

Os materiais podem ser agrupados por intermédio de uma curva, conhecida como a curva ABC, apresentado na Figura 82, abaixo. A curva ABC agrupa os materiais em percentual de valorização do estoque, em função do percentual de sua quantidade. Desta forma torna-se possível ao administrador agrupar os materiais de acordo com o seu valor, variando dos de maior valor agregado (materiais A) aos de menos valor (materiais C).

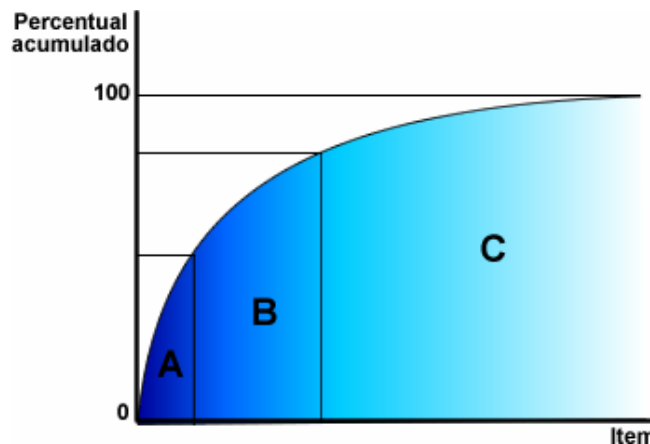


Figura 82 - Representação da Curva ABC, extraído de [MAR2006].

3. Quanto à importância operacional: Trata-se de agrupar materiais que podem prejudicar de maneira letal o processo operacional de uma empresa. Sua falta poderá acarretar sérios danos a um determinado setor e afetar indiretamente a outros setores e órgãos como, por exemplo, riscos de poluição ambiental, segurança industrial e até mesmo a vidas de pessoas. O custo da falta se torna mais oneroso do que o custo do investimento em estoque. Vale a pena adotar a classificação da importância operacional, para identificar quais são os materiais de grande valor operacional.

A classificação de materiais de importância operacional é conhecida também como *classificação XYZ* em que:

- Materiais X: consistem em materiais de importância mínima, que podem ser substituídos por um outro de natureza similar;
- Materiais Y: compreendem materiais de importância média, que podem possuir um similar, no entanto sua falta interfere na qualidade do serviço;

- Materiais Z: são as classes de materiais singulares, insubstituíveis. Sua falta acarreta a paralisação de um setor ou de uma ou mais fases operacionais.

B.5. Modelagem matemática da gestão de materiais: O ressuprimento

Ao falarmos sobre o processo de obtenção de um material, nos deparamos com duas questões singulares:

- **Quando** comprar o material?
- **Quanto** comprar de material?

Ambas as questões abrem uma antiga chaga que assola o estoque e que, em várias ocasiões, é passada por despercebida - o custo. Certamente existe um custo adicional pertinente quando o material não é requisitado no tempo e / ou na quantidade. Entretanto, estas questões podem ser respondidas através dos modelos matemáticos de ressuprimento.

A Figura 83 é conhecida como *curva dente de serra*. Esta é uma das curvas mais conhecidas no processo de gestão. Ela ilustra perfeitamente os efeitos da variação do estoque e os movimentos de entrada e saída de materiais em um sistema gestor. A *curva dente de serra* é composta por duas variáveis: “**tempo decorrido**” em função do “**consumo em unidades**”, representado pelo **saldo em estoque**.

As questões relativas a quando e quanto comprar podem ser respondidas a partir deste gráfico. Todavia, se faz necessário conhecer alguns elementos importantes que compõem a modelagem matemática de gestão dos materiais.

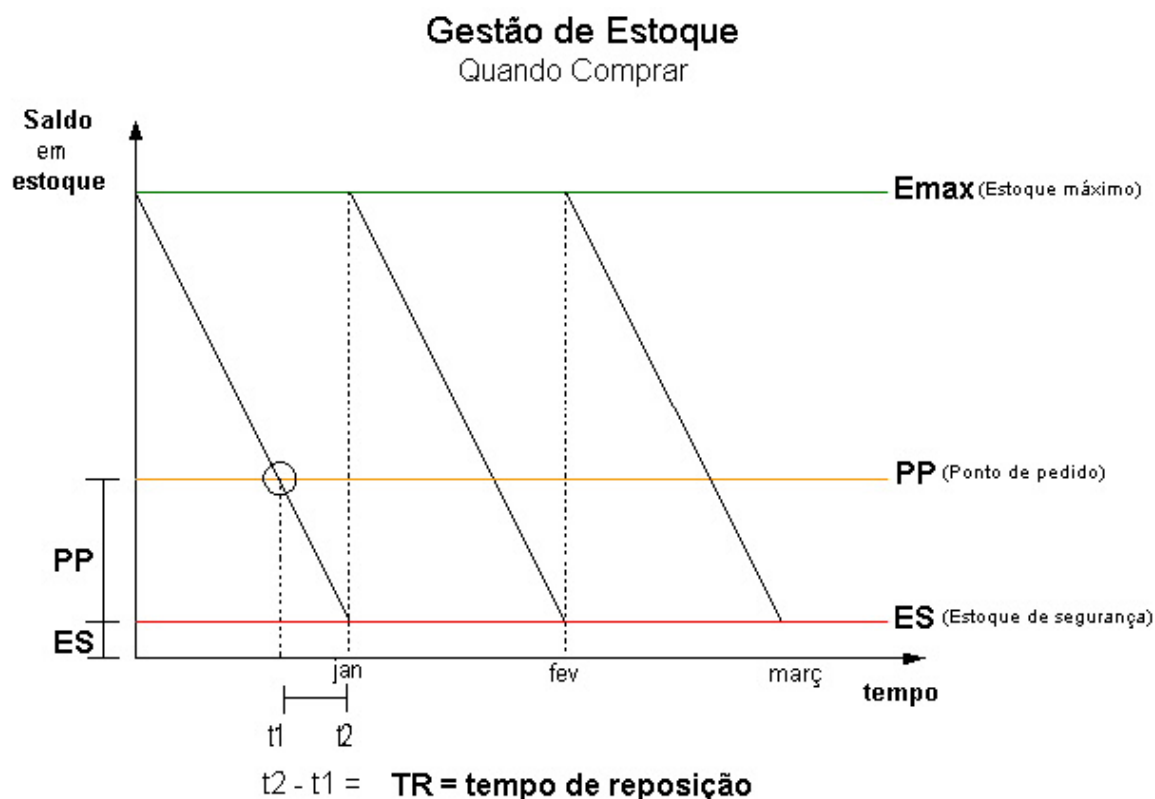


Figura 83 - Representação do gráfico dente de serra, extraído de [VIN2006].

B.5.1. Estoque Máximo (EMx) e Estoque Virtual (EV)

Representa a quantidade máxima permitida em estoque para um material em uma empresa. O *estoque máximo* é igual à soma do estoque de segurança mais o lote de compra, comentados neste capítulo. O lote de compra pode ser econômico ou não, apresentado mais a seguir.

$$EMx = ES + LC \quad (16)$$

O *EMx* também pode ser obtido através do *estoque virtual (EV)*. O EV é conhecido também como estoque disponível, é composto pelo *estoque real (ER)*, acrescido das quantidades das encomendas em andamento, conforme a fórmula abaixo:

$$EV = ER + Encomendas \quad (17)$$

Geralmente, algumas empresas acrescentam à fórmula do EV os materiais que estão em estado de inspeção (*Inspecao*). Deste modo, o EV assume a seguinte fórmula:

$$EV = ER + Encomendas + Inspecao \quad (18),$$

B.5.2. Estoque Mínimo ou de Segurança

Os modelos matemáticos de ressurgimento têm por função realizar a otimização do gerenciamento do estoque e também evitar a sua ruptura (atingir o nível zero), através da programação do estoque, de forma que haja uma reserva conhecida como *estoque de segurança* (ES). O ES, conhecido também como *estoque mínimo* (EMin), é uma quantidade de material excedente previamente calculada, destinada a atender uma situação de emergência, quando ocorrer um certo atraso no reabastecimento.

Os atrasos no reabastecimento podem ser ocasionados pela (o): oscilação no consumo; oscilação nas compras de aquisição (atraso no tempo de reposição); variação na qualidade, a qual ocorre quando o departamento de Controle de Qualidade rejeita um lote; fornecedor que envia um material divergente do solicitado; diferenças de inventário.

Quando o nível do estoque atinge o ES, isto indica que o determinado material está em estado crítico e seu estoque está próximo do risco de ruptura. O ES é obtido através da seguinte fórmula:

$$ES = K.TR.CM \quad (19)$$

onde o *K* é o *Fator de Segurança*, o *TR* representa o *Tempo de Reposição* e o *CM* é o *Consumo Médio*.

Fator de Segurança K

O *Fator de Segurança K* é uma medida em que são atribuídos níveis de tratamento para os materiais com maior grau de importância no estoque em situações críticas, como crises periódicas de fornecimento e consumo imprevisível. Ele foi modelado em função da importância operacional e do valor do material, ou seja, através dos materiais classificados pelos modelos de classificações XYZ e ABC em que K é obtido de acordo com a distribuição normal de probabilidades. Na Tabela 4, é apresentado o fator de segurança K, de acordo com as classificações ABC e XYZ dos materiais.

Importância Operacional	Valor de Consumo	K
Z	A	0,5
	B	0,7
	C	0,9
Y	A	0,3
	B	0,4
	C	0,8
X	A	0,1
	B	0,2
	C	0,6

Tabela 4 – Tabela do Fator de Segurança (K), extraído de [VIN2006].

B.5.3. Tempo de Reposição

Para realizar o cálculo do ES, é necessário que se tenha a estimativa do *tempo de reposição* (TR). O TR consiste no intervalo de tempo estimado desde a verificação de que o estoque necessita ser repostado à sua efetiva chegada no almoxarifado. O TR é composto por três fases:

- T1 - tempo de emissão do pedido de compra até a sua chegada ao fornecedor;

- T2 – tempo gasto para preparação do pedido pelo fornecedor para deixá-lo em condição de ser transportado, ou seja, tempo para fabricar os produtos, separar os produtos, emitir fatura, etc;
- T3 – tempo de transporte entre o fornecedor até o seu recebimento pelo almoxarifado.

Tecnicamente, o tempo de reposição pode ser composto por mais fases e cabe ao responsável competente defini-las. Quanto mais fases o TR tiver, mais precisa e realista será a sua estimativa. Deve-se tomar um certo cuidado com a estimativa do TR pois, se estimado indevidamente, ele pode distorcer toda a estrutura de planejamento do estoque.

B.5.4. Quando Comprar (QC)

Conhecidos os conceitos básicos sobre a gestão de estoque, é possível inferir sobre o processo de compra de materiais.

A resposta à questão do “quando comprar” pode ser obtida conhecendo, basicamente, o sistema de reposição adotado pela empresa. Existem dois tipos de sistema de reposição: Periódico e Contínuo. O *sistema de reposição periódico*, conhecido também como sistema de reposição em períodos fixos, trata-se de um sistema de reposição de natureza periódica, ou seja, possui um intervalo padrão de reposição e depois que é atingido um tempo pré-estabelecido, um novo pedido é realizado. O *sistema de reposição contínua*, também conhecido como sistema de ponto de pedido ou sistema de lote padrão, parte do princípio que o processo de compra é disparado quando o nível do estoque atinge uma determinada marcação, conhecida como PP (*ponto de pedido*).

B.5.5. Ponto de Pedido (PP)

O ponto de pedido, conhecido também como ponto de ressuprimento, é uma marcação no estoque. Indica que quando o saldo do estoque de um devido material estiver abaixo desta marcação, terá que ser repostado. O *PP* pode ser calculado pela seguinte fórmula:

$$PP = CM \cdot TR + ES \quad (20)$$

onde *TR* representa o *Tempo de Reposição*, *ES* é o *Estoque de Segurança* e *CM* é o *Consumo Médio*.

O **CM** utilizado para calcular o ponto de pedido pode ser:

- Mensal: quando o ciclo de gestão for semestral ou anual;
- Diário: quando o ciclo de gestão estimado for mensal;

B.5.6. Quanto Comprar

A resposta à questão de “quanto repor” de material pode ser obtida através da forma tradicional de compra, ou pelo lote de compra. Nesta forma específica para determinar a quantidade a ser comprada, verifica-se a quantidade disponível no estoque (*EV*) menos a quantidade que falta para atingir o *Estoque Máximo* (*Emx*), denotado pela fórmula abaixo.

$$QC = Emx - EV \quad (21)$$

B.5.6.1. Lotes de Compra

O lote de compra trata-se do valor ideal de material a ser adquirido para suprir uma determinada demanda. O lote pode ser econômico ou não, de acordo com Martins [MAR2006].

- **Lote Econômico de Compra**

O *Lote Econômico de Compra* (LEC) visa adquirir um lote ideal para consumo. Tem por propósito evitar ao máximo gastos com armazenamento, obtenção e manutenção. O LEC está denotado pela seguinte fórmula:

$$LEC = \sqrt{2.CA.CC} / CPA.PU \quad (22)$$

onde:

CA = consumo anual em quantidades;

CC = custo unitário do pedido de compra;

CPA = custo do material armazenado;

PU = preço unitário do material.

No entanto, existem algumas limitações com relação ao uso do LEC. O modelo não é sensível à variação da quantidade no lote [VIN2006]. Caso o lote sofra uma variação na quantidade entre o lote adquirido e o econômico, o seu custo total sofre pouca variação. Na prática, este é um fator decisivo para a escolha do modelo de lote de compra, pois, nas situações reais, freqüentemente ocorre a variação da quantidade, bem como os prazos de entrega do material. Em conformidade com Viana [VIN2006] e Bowersox [BOW1999], o LEC pode ser utilizado como um simulador e não como um otimizador do estoque. Por estas razões, vale apenas utilizar como sistema de aquisição de materiais, o LCC (Lote de Compra Orientado a Consumo).

- **Lote de Compra Orientado a Consumo (LCC)**

O *Lote de Compra orientado a Consumo* (LCC) é um dos métodos mais tradicionais para obtenção de material por uma empresa. Nas situações ideais, o LCC é calculado com base no consumo médio do material, em função de uma *política de compra* (w). A política de

compra consiste no período de tempo desejado de cobertura do estoque. O LCC pode ser obtido pela seguinte fórmula:

$$LCC = w.CM \quad (23),$$

onde o CM representa o *Consumo Médio* e w é a *Política de Compra*.

A política de compra w , representa o tempo de duração do estoque em meses de consumo, podendo assumir valores de 1, 2, 3, 6, até 12 de meses de consumo.

B.6. Síntese do capítulo

Este capítulo abordou alguns conceitos básicos utilizados no mundo da administração de materiais, mais especificamente sobre a gestão de materiais, definições de estoque, classificação de materiais em nível de importância e valor acumulado e algumas modelagens matemáticas, inerentes ao tempo correto para fazer a requisição de compra de material e sua quantidade necessária.

B.7. Bibliografia

[ARA1975] Araújo, J. S. *Administração de Materiais*. 4º Ed. São Paulo – SP, 1975. Editora: Atlas.

[BOW1999] Bowersox, D. J., Closs D. *Logistical management: the integrated supply chain process*. Michigan State University. Michigan, EUA, 1996. Ed. McGraw-Hill Companies, Inc.

[COS2002] Costa, F. J.C. L. *Introdução à Administração de Materiais em Sistemas Informatizados*. 1ª Ed. São Paulo – SP, 2002. Editora: Editco Comercial Ltda.

[DIA1985] Dias, M. A. P. *Administração de Materiais*. Ed. Compacta, São Paulo – SP, 1985. Editora: Atlas.

[DOR1981] Dorn, W. S., McCracken, D. D. *Cálculo numérico com estudos de casos em FORTRAN IV*. Rio de Janeiro Editora: Campus, São Paulo, 1981. Ed. Universidade de São Paulo.

[MAR1999] Martin, C. *Logística e Gerenciamento da Cadeia de Suprimentos: Estratégia para a Redução de Custos e Melhoria dos Serviços*, 1ª Ed. Tradução: Francisco Roque Monteiro Leite. São Paulo – SP, 1999. Editora: Pioneira.

[MAR2006] Martins, P.G.; Campos, P. R. *Administração de Materiais e Recursos Patrimoniais*. 2º Ed. São Paulo – SP, 2006. Editora: Saraiva.

[SPI1977] Spiegel, M. R. *Estatística: resumo da teoria, 875 problemas resolvidos, 619 problemas propostos*. São Paulo - SP, 1977. Editora: McGraw-Hill do Brasil.

[VIN2006] Viana, J. J. *Administração de Materiais: Um Enfoque Prático*. 1º Ed., São Paulo – SP, 2006. Editora: Atlas.

Anexo C - Modelagem UML das ações e percepções dos elementos da sociedade de Tutores Inteligentes

C.1. Diagrama de Classes dos Principais processos de ação do AGT

Nesta seção, são apresentados os diagramas de classes dos principais processos de ação do AGT, enfatizados no capítulo 4 (seção As Ações do AGT) pelo caso de uso de ações.

C.1.1. Diagrama de classe Calcular a Lista de Materiais que irão Exceder e Zerar

A Figura 84 representa o diagrama de classe da ação “Calcular a Lista de Materiais que irão Exceder e Zerar” no estoque. Neste diagrama, o AGT invoca a classe gerenciadora *Material_Excedentes_Zerados_Generator*, a qual é responsável por calcular a lista de materiais excedente e zerados e também armazenar o resultado de seu processamento na tabela *Material_Excedentes_Zerados* do banco de dados do AGT. Os dados necessários para o processamento são recuperados pela classe *Gestão_Estoque*.

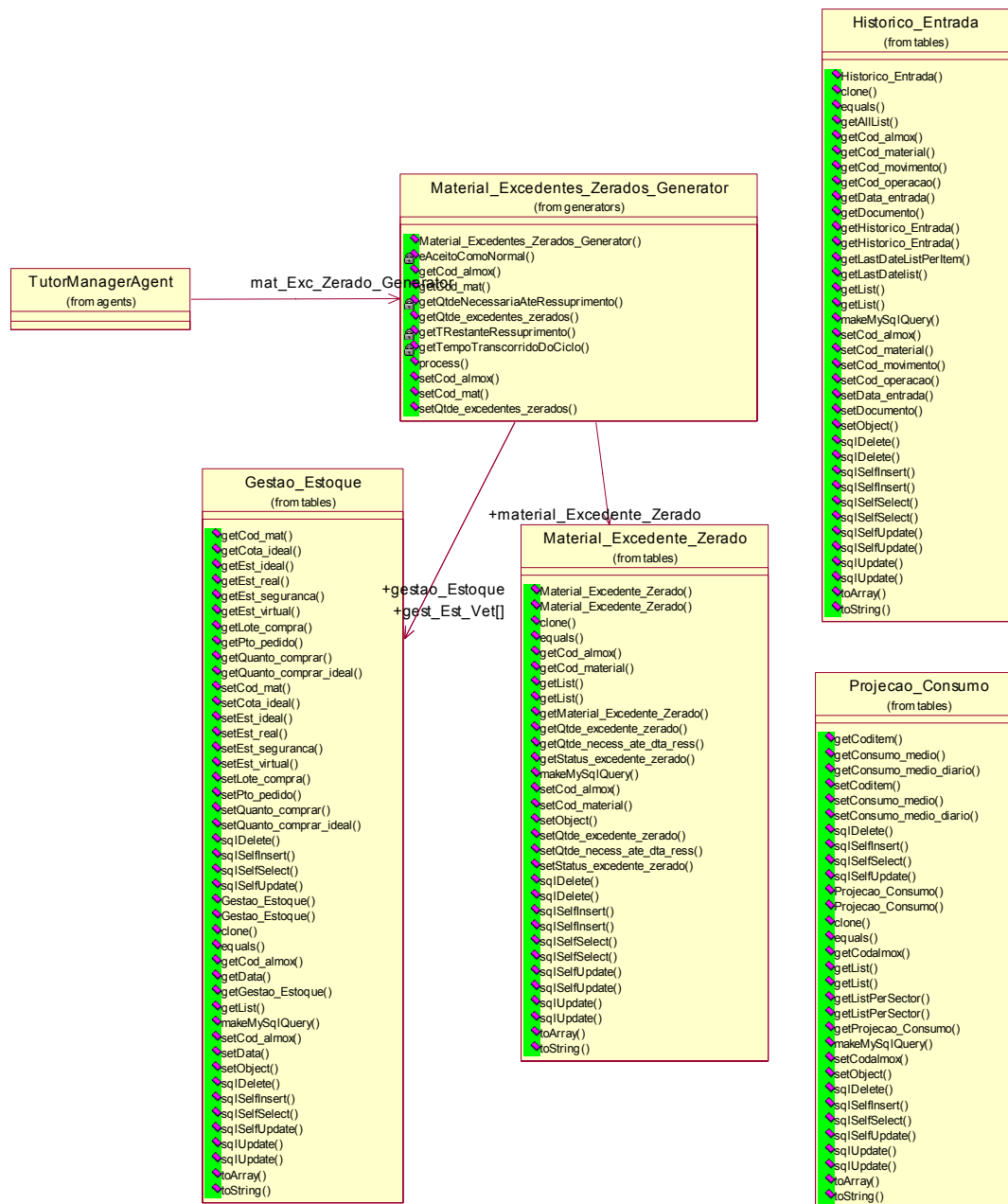


Figura 84 - Diagrama de classe Calcular a Lista de Materiais que irão Exceder e Zera.

C.1.2. Diagrama de classe Calcular o Estado Geral do Estoque

A Figura 85 representa o Diagrama de classe *Calcular o Estado Geral do Estoque*. Esta ação é feita pela classe *Estado_Geral_Estoque_Generator_Nova*, invocada pela classe do AGT *TutorManagerAgent*. A classe *Estado_Geral_Estoque_Generator_Nova* recupera a lista de materiais excedentes e zerados e de materiais que possuem lotes para vencer. Esta classe calcula o nível de risco de cada parâmetro do estado geral e armazena na tabela *Estado_Geral_Estoque*, da base de dados do AGT, conforme comentado no capítulo 4.

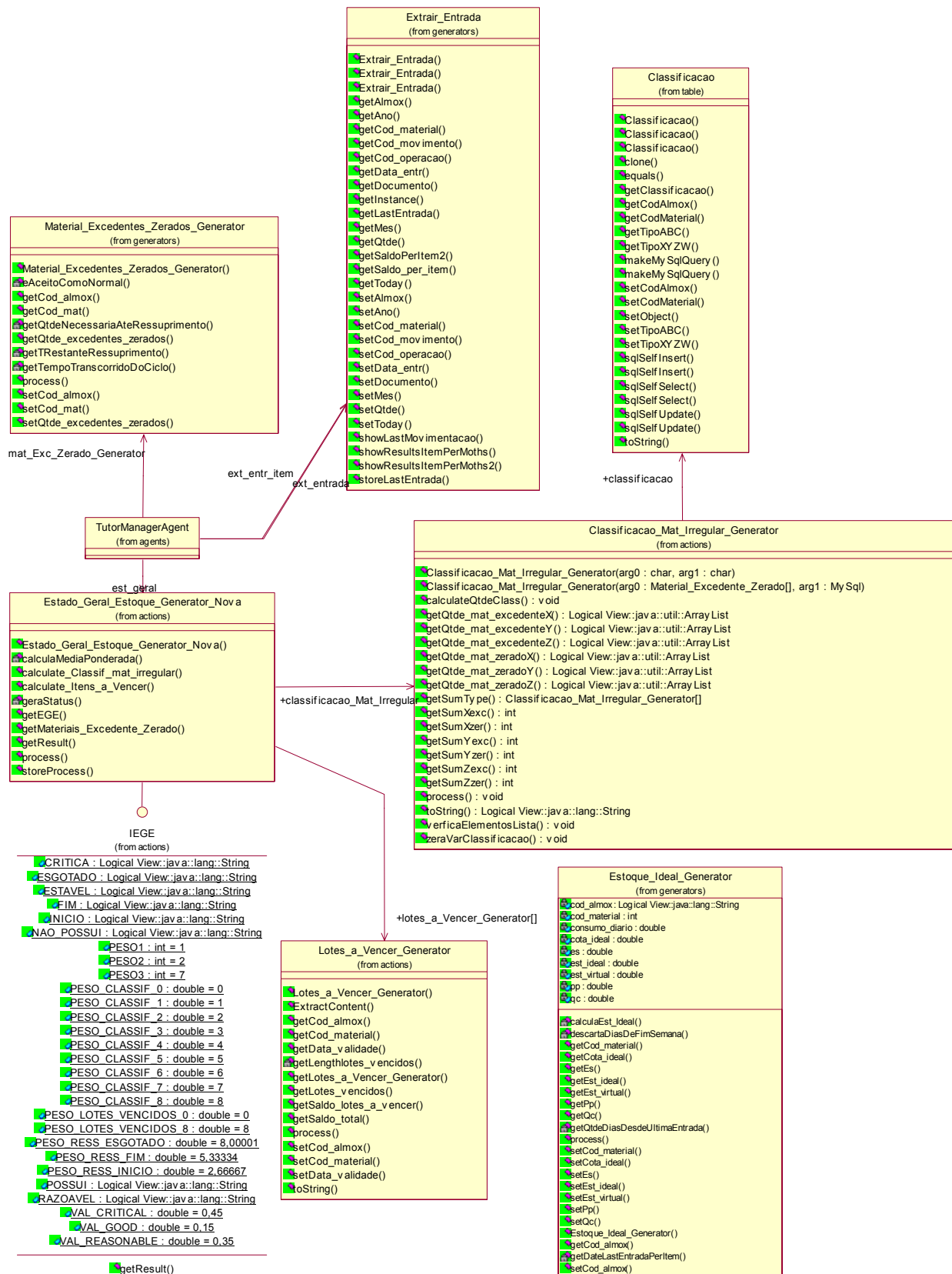


Figura 85 - Diagrama de classe faz os cálculos de Estado Geral do Estoque.

C.1.3. Diagramas de classe de entrar na sociedade local e remota

O diagrama de classe da Figura 86 efetua a entrada na sociedades Local e Remota. Este processo é disparado pelo método *enterSociety()* da classe *TutorManagerAgent*. A classe *TutorManagerAgent* contém os atributos privados *msg_local* e *msg_remote*, os quais constituem a associação com a classe *MBoxSag*. A classe *MBoxSag* contém métodos para criar sociedades e realizar a entrada de agentes nas sociedades geradas.

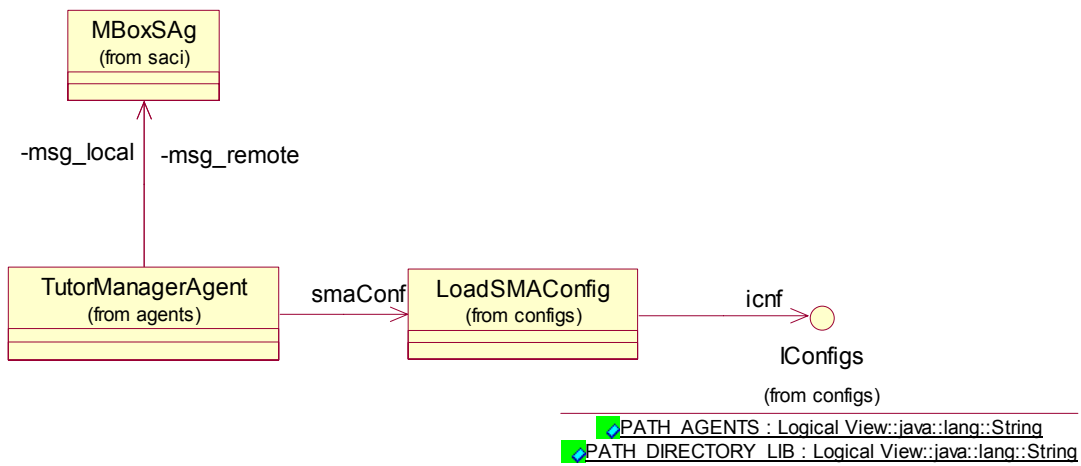


Figura 86 - Diagrama de classe - Entrar na sociedades Local e Remota.

C.2. Diagramas de subcasos de uso e diagramas de classe das ações do AGT

A Figura 87 contém o caso de uso de ação do AGT, que recupera a última lista de suprimento de materiais, vinda do almoxarifado central na base de dados do Sig2m da UBS.

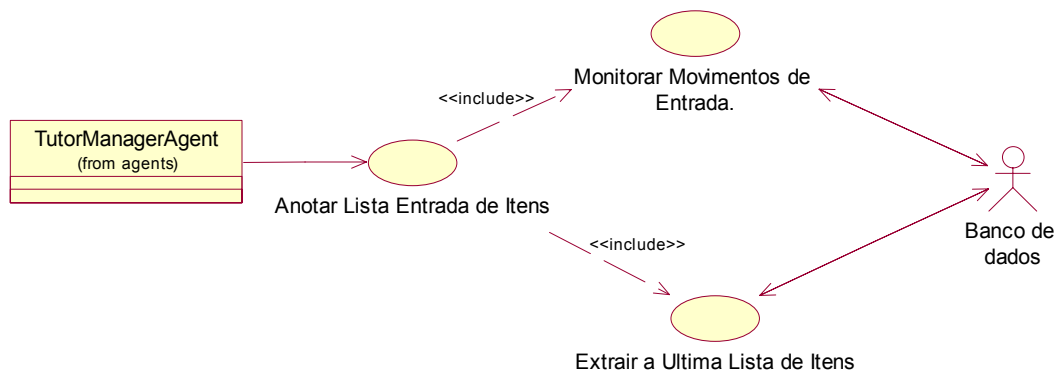


Figura 87 - Caso de uso que recupera a lista de suprimento de materiais vindas do almoxarifado central.

O diagrama de classe do processo “Anotar Lista de Entrada de Itens” é apresentado pela Figura 88. O AGT associa-se com a classe *Extrair_Entrada*, que contém métodos que recuperam a última lista de suprimento, encontrada na tabela de movimentações do Sig2m. Os materiais da última lista de suprimento estão armazenados em uma tabela da base de dados do AGT, chamada *Histórico_Entrada*.

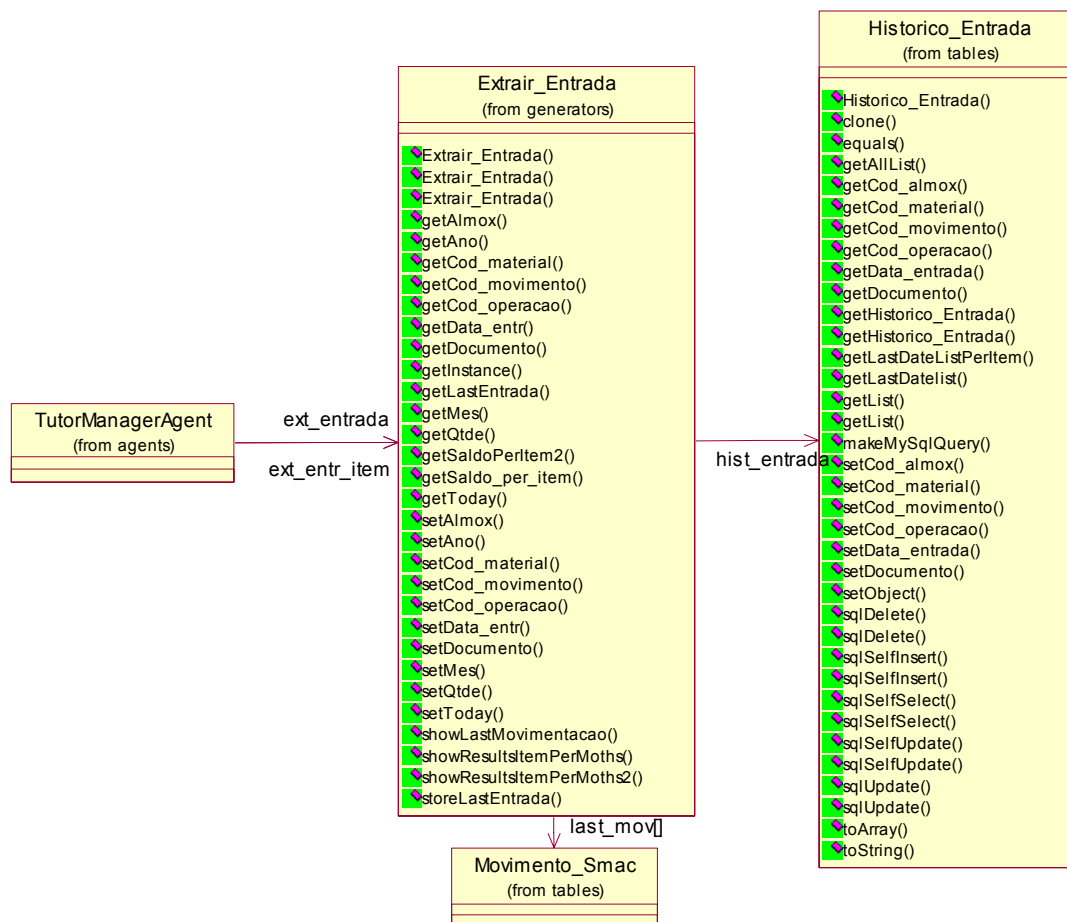


Figura 88 - Diagrama de classe Anotar Lista Entrada de Itens.

O caso de uso da Figura 89 faz a análise do nível do Estado Geral do Estoque e de cada parâmetro que o compõe, que são: materiais excedentes, zerados e materiais com lotes em véspera de vencimento.

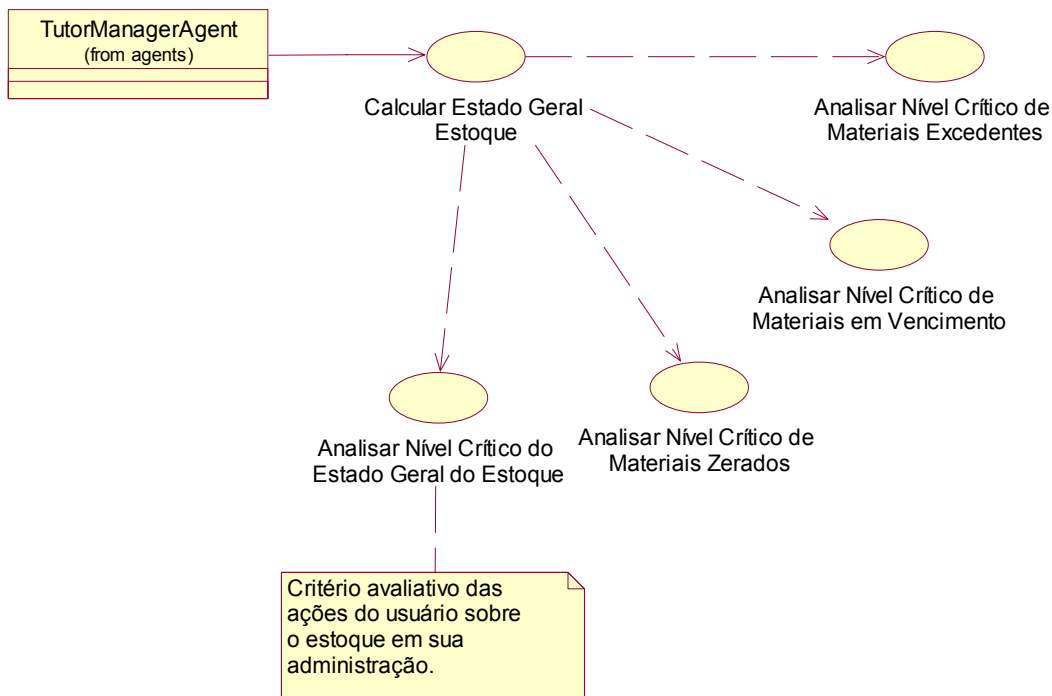


Figura 89 - Caso de uso de análise do Estado Geral do Estoque.

O caso de uso da Figura 90 faz a comunicação com a sociedade remota para requisitar e doar materiais. O processo “Comunicar com outros *TutorManagerAgent*” faz o envio e o recebimento de mensagens KQML, vinda dos membros da sociedade remota (AGTs). O processo “Enviar Mensagens Remotas KQML” contém as ações de montar o conteúdo da mensagem e definir o seu tipo de envio. O envio de mensagens remotas pode assumir três tipos:

- Requisição de Materiais: mensagem de requisição de materiais aos setores;
- Proposta de doação de matérias: mensagem de proposta à doação de materiais para um setor requisitante;
- Aceitação e / ou Rejeição de doação de materiais: mensagem de aceitação e / ou rejeição de uma proposta de doação de materiais.

O processo “Receber Mensagens Remotas KQML” faz o recebimento de mensagens de outros AGTs. O conteúdo da mensagem é extraído e armazenado no banco de dados do AGT. As mensagens que chegam da sociedade remota possuem três tipos:

- [illegible]

Figura 90 - Caso de uso de comunicação com a sociedade remota.

Na Figura 91 é apresentado um caso de uso, cujo AGT informa o usuário sobre a chegada de mensagens provenientes da sociedade remota. O AGT verifica em sua base de dados sobre a chegada de mensagens e, caso tenha, ele monta uma mensagem KQML e a envia ao Agente Mensageiro (AM) para informar o usuário sobre a chegada destas mensagens.

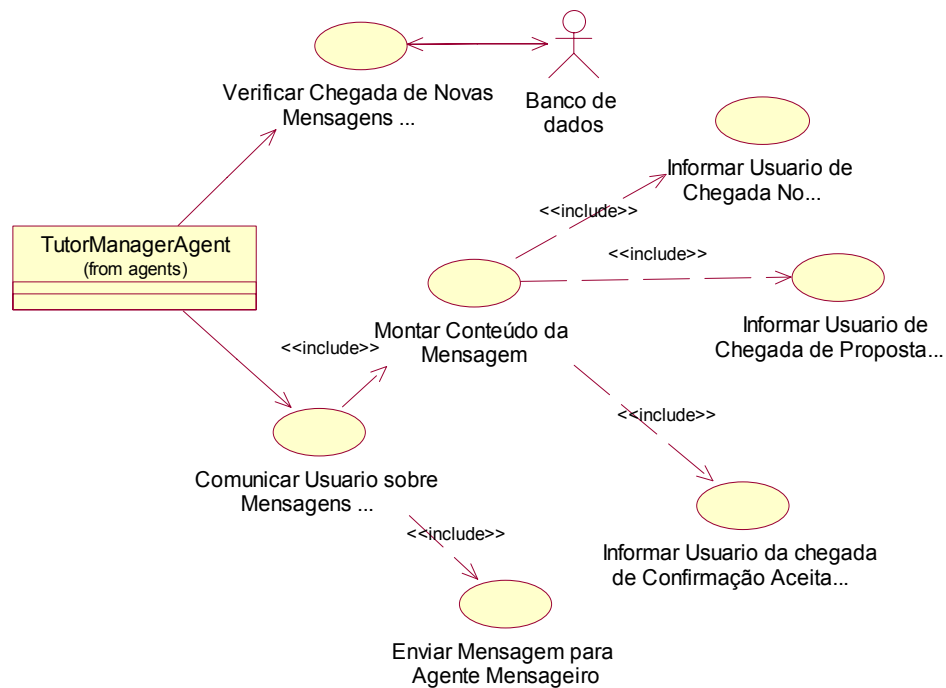


Figura 91 - Caso de uso das ações de informar o usuário sobre a chegada mensagens remotas.

O diagrama de caso de uso, apresentado pela Figura 92, faz os cálculos de gestão de estoque. O processo de gestão de estoque é composto pelos processos de histórico de consumo, projeção de consumo, estoque real e virtual, estoque de segurança, lote de compra, ponto de pedido, quanto comprar e estoque ideal.

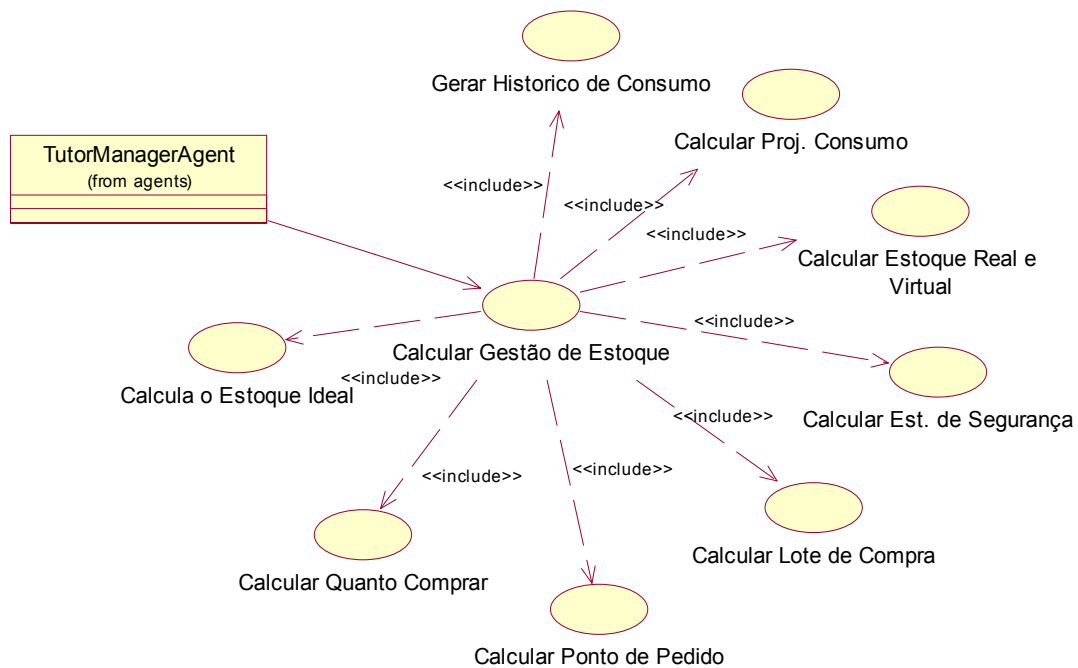


Figura 92 - Caso de uso de calcular Gestão do Estoque.

O cálculo de *Gestão de Estoque* é feito pelas classes apresentadas pela Figura 93, em que a classe de *Gestão_Estoque_Generator* é responsável por gerenciar e recuperar o resultado do processamento das classes geradoras:

- Estoque_Real_Virtual_Generator;
- Lote_Compra_Generator;
- Sazonalidade_Generator;
- Ponto_Pedido_Generator;
- Quanto_Comprar_Generator;
- Estoque_Seguranca_Generator;
- Estoque_Ideal_Generator.

Cada classe geradora é responsável por calcular uma variável inerente a gestão.

Na Figura 94, é apresentado o caso de uso que orienta o usuário sobre o estado geral do estoque. A orientação do usuário é feita através das mensagens que são enviadas pelo AGT à sociedade local. A priori, é montado o conteúdo da mensagem e em seguida é enviada, através do AM.

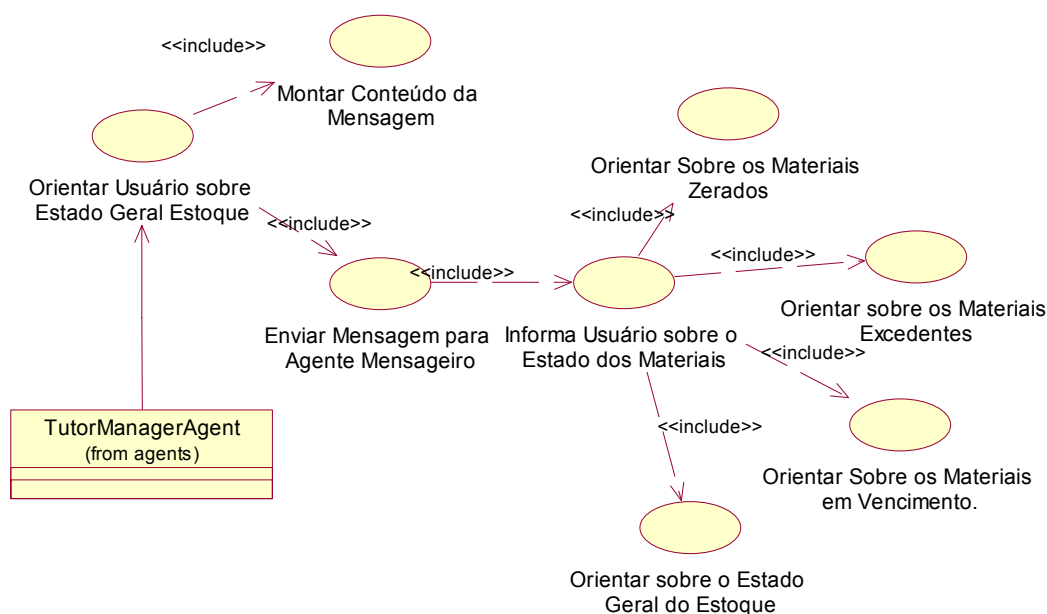


Figura 94 - Caso de uso para orientar o usuário sobre o estado geral do estoque.

C.3. Diagrama de Classes dos Principais processos de percepção do AGT

Esta seção apresenta os diagramas de classes dos principais processos de percepção do AGT. As classes da Figura 95 fazem a verificação sobre a existência de novas requisições de materiais, feitas pelo usuário da UBS para ser enviada para a sociedade remota. No processo de requisição de materiais, o usuário da UBS faz a sua requisição, a qual é armazenada no banco de dados. A verificação de novas requisições é feita quando a classe *TutorManagerAgent* adquire uma crença de verificar novas requisições na base de dados. Então é disparado o processo de busca por requisições realizadas pela classe

NewRequests. A classe *BrfManager* administra a adição de uma crença por vez, possibilitando a execução de um plano até o seu fim ou fracasso.

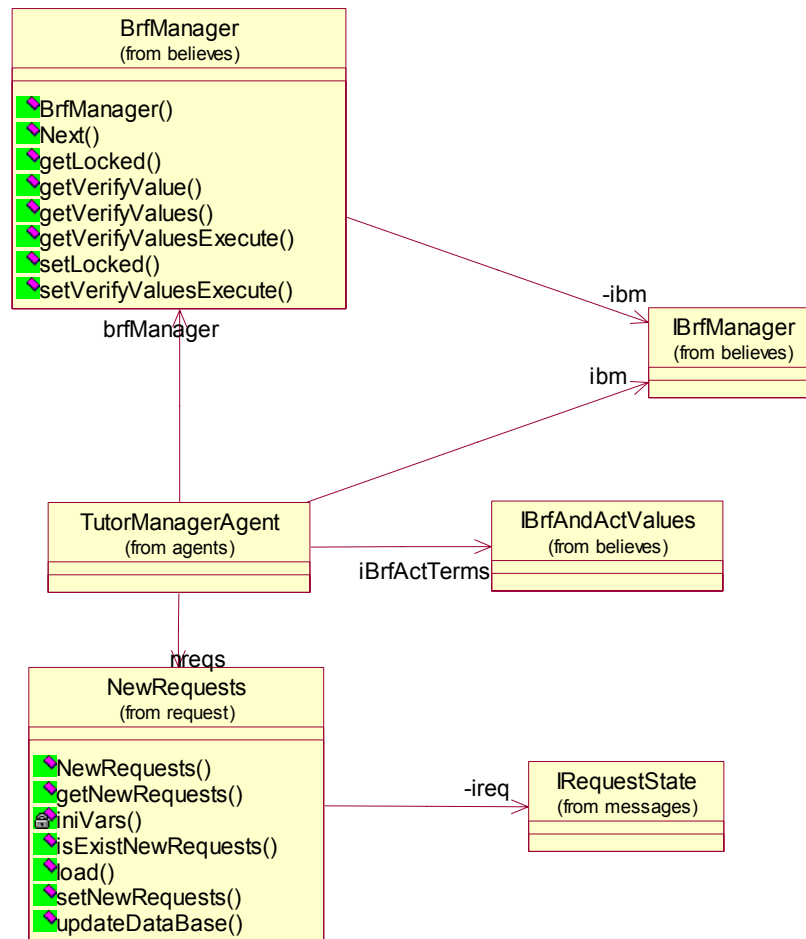


Figura 95 - Diagrama de classe - Verificar se Existe Novas Requisições para Enviar para os Setores.

Anexo D – Mensagens KQML do Protocolo de Negociação de Requisição e Doação de Medicamentos, realizado pela Sociedade Remota

Quando uma UBS necessitar de medicamentos durante um ciclo de distribuição, o usuário gera uma lista de requisição de itens, a ser enviada para as demais UBSs. Então o AGT requisitante encaminha a lista de requisições a todos os outros demais AGTs da sociedade remota, candidatos à doação, intermediado pelo agente facilitador, passos 1 e 2 da Figura 96. O AGT doador, percebe a chegada da lista de requisição de medicamentos, armazena-a em sua base dados e realiza a ação de informar o usuário sobre a sua chegada. Se a UBS, candidata à doação, possui itens excedentes, seu usuário pode gerar uma doação do material para o setor requisitante. A partir da lista de requisição, o usuário faz a sua proposta de doação de medicamentos e o AGT doador realiza ação de enviar a proposta de doação gerada para o AGT requisitante e espera a sua confirmação de aceitação e / ou rejeição (passo 3 da Figura 96).

Ao receber a lista de proposta de doação na UBS requisitante, o AGT informa o usuário sobre a sua chegada. O usuário, por sua vez, analisa a proposta de doação e confirma a sua aceitação e / ou rejeição (passo 4), que é novamente encaminhada para o respectivo setor doador para gerar a sua efetiva saída.

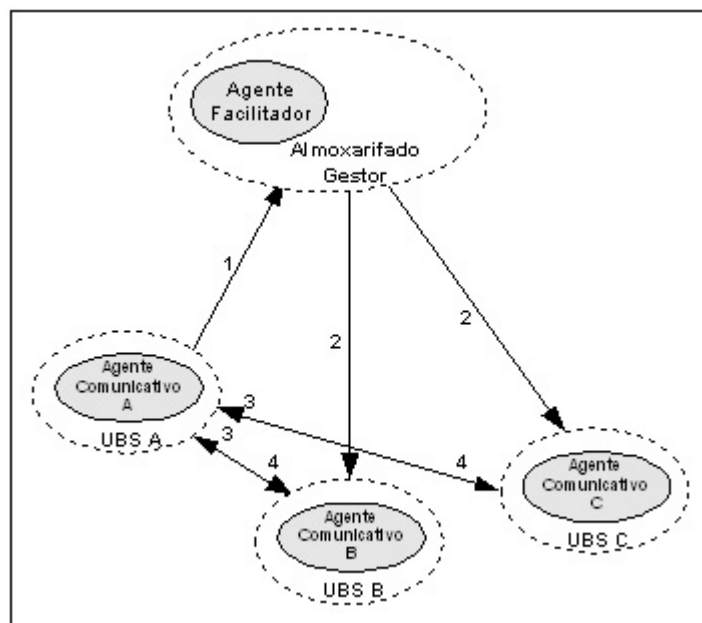


Figura 96 – Detalhes do processo de requisição e doação de medicamentos entre UBSs.

D.1. Mensagens KQML enviadas pelos agentes comunicativos de cada UBS

A seguir, serão apresentadas as mensagens KQML, geradas pela sociedade remota, a fim de realizar o processo de requisição e doação de materiais entre UBSs, apresentada pela Figura 96.

Requisição de Material – UBS A

No processo de requisição de materiais, a UBS A envia uma mensagem de requisição de materiais para a sociedade remota, através do agente facilitador que a encaminha para todos os agentes da sociedade (passo 1 e 2 da Figura 96). Na Tabela 5, é apresentada a mensagem KQML, de requisição de materiais. O componente *content* é um objeto serializável¹¹, que armazena objetos de requisição de material, que contêm atributos como o

¹¹ Objeto serializável: tipo de objeto dotado da capacidade de ser transmitido em uma conexão de rede.

código do material, descrição e a quantidade a ser requisitada. A ontologia *supply* é necessária para que os agentes possam reconhecer informações sobre o modelo de domínio para se comunicarem.

ask-one :reply-with :receiver SOCIETY_REMOTE :ontology supply :saci-object-stream-content yes :sender S00012 :content serializedObject :language objects

Tabela 5 - Mensagem remota KQML de requisição de material enviada a sociedade remota através do agente facilitador.

Atendimento de uma requisição

O AGT B da UBS B obtém a percepção da chegada de uma requisição de materiais enviada pela UBS A. Ele a armazena em seu banco de dados e adota a ação de informar o usuário do setor sobre a sua chegada, enviando uma mensagem local KQML para o Sig2m, apresentada pela Tabela 6. Note que a estrutura mensagem local o *parâmetro* :language indica que o conteúdo da mensagem trata-se de um texto em português.

tell :reply-with :receiver LocalAgent :ontology communication :saci-object-stream-content yes :sender 023787 :content serializedObject :language portuguese_text

Tabela 6 - Mensagem local KQML enviada pelo AGT B ao Sig2m da UBS B sobre a chegada de uma requisição de material.

As mensagens de chegada de requisição são apresentadas ao usuário através de uma caixa de diálogo. O usuário do setor entra em uma tela do Sig2m, em que é possível verificar as requisições de materiais, pelas quais é gerada uma proposta de doação dos materiais requisitados. A proposta de doação é armazenada em uma base de dados e o AGT obtém a percepção de sua existência e adota a ação de enviá-la através de uma mensagem remota ao setor requisitante (UBS A), vide Tabela 7.

tell :reply-with :receiver S00012 :ontology supply :saci-object-stream-content yes :sender 023787 :content serializedObject :language objects

Tabela 7 - Mensagem remota KQML de proposta de doação de medicamentos emitida pelo AGT B ao AGT A.

A UBS C segue a mesma estratégia que a UBS B. Ao receber uma mensagem remota de requisição vinda da UBS A, o AGT C armazena-a e envia uma mensagem local ao Sig2m para informar o usuário sobre a sua chegada (apresentado pela Tabela 8).

<pre>tell :reply-with :receiver LocalAgent :ontology communication :saci-object-stream-content yes :sender 024457 :content serializedObject :language portuguese_text</pre>

Tabela 8 - Mensagem local KQML enviada pelo AGT C ao Sig2m da UBS C sobre a chegada de uma requisição de material.

O usuário, por sua vez, faz uma proposta de doação de materiais e o AGT realiza a ação de enviá-la através de uma mensagem KQML ao setor requisitante (UBS A), apresentado pela Tabela 9.

<pre>tell :reply-with :receiver S00012 :ontology supply :saci-object-stream-content yes :sender 024457 :content serializedObject :language objects</pre>

Tabela 9 - Mensagem remota KQML de proposta de doação de medicamentos emitida pelo AGT C ao AGT A.

Confirmação do requisitante sobre a aceitação da doação de materiais

Ao receber a proposta de doação vinda dos setores, o AGT A armazena em seu banco de dados e realiza a ação de informar o usuário, enviando uma mensagem local ao Sig2m, conforme a Tabela 10.

<pre>tell :reply-with :receiver LocalAgent :ontology communication :saci-object-stream-content yes :sender S00012 :content serializedObject :language portuguese_text</pre>

Tabela 10 - Mensagem local KQML enviada pelo AGT A ao Sig2m da UBS A sobre a chegada de proposta de doação de materiais.

No Sig2m, o usuário seleciona a proposta de doação que mais lhe favorece, gerando uma confirmação de aceitação da proposta de doação dos setores. O AGT realiza a percepção sobre a decisão do usuário e gera a ação de enviar a confirmação às UBSs B e C da sociedade, sob forma de mensagem KQML, representadas, respectivamente, pelas tabelas:

Tabela 11 e Tabela 12. O processo de negociação de proposta de doação de materiais atinge o estado final quando as UBSs doadoras realizam efetivas saídas do materiais para o setor requisitante.

tell :reply-with :receiver 023787 :ontology supply :saci-object-stream-content yes :sender S00012 :content serializedObject :language objects

Tabela 11 - Envio da mensagem remota de confirmação de aceitação da doação de materiais para a UBS B.

tell :reply-with :receiver 024457 :ontology supply :saci-object-stream-content yes :sender S00012 :content serializedObject :language objects

Tabela 12 - Envio da mensagem remota de confirmação de aceitação da doação de materiais para a UBS C.

As mensagens KQML, enviadas entre os AGTs, possuem como *content* objetos que armazenam informações dos materiais requisitados e doados. Os objetos enviados são basicamente três:

- *Requisicao_Material*: objeto que contém informações sobre a requisição de um material;
- *Atendimento_Requisicao*: objeto que detém informações sobre o atendimento de uma requisição;
- *Proposta_Doacao*: objeto que contém informações inerentes à proposta de doação de materiais.

**Anexo E – Artigo apresentado no CBIS - X
Congresso Brasileiro de Informática em Saúde
"Informática em Saúde e Cidadania".
Florianópolis – SC**

**Uma Proposta de Sociedade de Agentes
Inteligentes para a Gestão de Medicamentos nas
Unidades Básicas de Saúde**

Ekler Paulino de Mattos, Leonardo de Souza Mendes, Maurício Bottoli

Faculdade de Engenharia Elétrica e Computação (FEEC),
Universidade Estadual de Campinas (Unicamp), Brasil

Resumo - O trabalho proposto tem por objetivo apresentar o uso da arquitetura de sistemas Multi-Agentes cognitivos como uma solução para o problema da distribuição de medicamentos nas unidades básicas de saúde (UBS). Este problema apresenta características para classificá-lo como sendo de natureza distribuída: problema genérico que pode ser dividido em subproblemas (cada UBS como parte do modelo de domínio). O uso de sistemas Multi-Agentes cognitivos podem auxiliar as UBSs a regularizarem a situação de seus estoques e obterem cotas de medicamentos mais próximas à sua realidade. Isto é possível através de procedimentos matemáticos de gestão de estoque.

É apresentado como resultado um estudo de caso onde é realizada a simulação de uma rede de UBSs que utiliza o sistema multiagente como assistente para a resolução de tal problema.

Palavras-chave: Inteligência Artificial Distribuída, Sistema Multi-Agentes Cognitivos, Gestão de Medicamentos.

Abstract - The goal of the proposed work is to present the use of the cognitive multi-agent systems's architecture as a solution for the problem of medicine distribution in the health basic units (HBU). This problem presents characteristics to be classified as distributed nature: generic problem that can be disposed in subproblems (each HBU as part of a domain model). The use of cognitive multi-agent systems may assist the HBUs to regularize the situation of its supplies and obtain medicine quotas nearer to its reality. That is possible through supply management's mathematical procedures.

It's introduced as result of a case study where is realized a HBU network's simulation that uses the multi-agent system as assistant for the resolution of such problem.

Key-words: Distributed Artificial intelligence, Cognitive MultiAgent Systems, Medicine Management.

Introdução

O processo de informatização nas redes municipais de saúde tem trazido uma série de contribuições para a melhoria na qualidade de vida do munícipe. A agilização do processo de atendimento ao público e a rapidez no processamento da informação coesa e precisa são bons exemplos a serem citados. Todavia, mesmo com a informatização e interconexão lógica das Unidades Básicas de Saúde (UBSs) uma série de deficiências ainda está presente, tal como a ausência e/ou o excesso de medicamentos para alguns setores. Estimar um estoque padrão de medicamentos, definir precisamente a quantidade necessária para atender um ciclo de distribuição (cota), prever períodos sazonais, entre outras, são tarefas complexas, que necessitam de mão-de-obra especializada, no caso de gestores de estoque (pouco comum nas redes municipais de saúde) e de sistemas computacionais que auxiliem na execução destas tarefas.

O problema de distribuição de materiais para as UBSs pode ser categorizado como um problema de natureza distribuída, ou seja, um problema complexo que não pode ser solucionado apenas por uma instância de software [1].

Por estas motivações, a arquitetura das redes municipais de saúde, torna-se palco ideal para o desenvolvimento de uma proposta de solução distribuída destinada a resolver a questão da distribuição de materiais, por meio do uso de procedimentos de gestão de estoque de medicamentos com Inteligência Artificial Distribuída (IAD), mais especificamente, com o uso de uma sociedade de agentes cognitivos.

Inteligência Artificial Distribuída (IAD)

A resolução de problemas distribuídos é o nome aplicado ao subcampo da inteligência artificial distribuída (IAD) em que enfatiza o uso de agentes para trabalharem juntos em problemas que requerem um esforço coletivo [2].

Basicamente um sistema de IAD é formado pelos seguintes elementos [3]: *Agentes* - entidades ativas do sistema; *Sociedade* - conjunto formado por agentes; *Ambiente* - conjunto formado pelos objetos do sistema (objetos passivos); *Interação* - troca de informações entre os agentes (podem ser trocas diretas através de comunicação

explícita, ou pelo ambiente) e *Organização* - garante que os agentes farão aquilo para o que foram projetados.

Agentes Inteligentes

Existem várias definições sobre agentes [3] [4] [5]. De forma genérica um agente é uma entidade (seja real ou não) que está inserida em um contexto (em um ambiente), o qual possui comportamento autônomo: perceber, agir, deliberar e comunicar-se com outros agentes de forma a alterar o estado do ambiente.

Tipos de Agentes

Os agentes são basicamente classificados em dois grupos: agentes reativos e agentes cognitivos. Os agentes reativos são mais simples, reagem instintivamente, formado pelo par Estímulo-Resposta (Ação-Reação). São baseados em modelos de organização biológica (como peixes, formigas e cupins). Ao contrário dos agentes reativos, os agentes cognitivos são mais complexos, dotados de estados mentais (como crenças, desejos, intenções) e funcionam racionalmente. Constroem um conjunto de planos e ações para atingir um objetivo pretendido.

A arquitetura de agentes fundamentada em estados mentais (Crenças, Desejos e Intenções)

Uma arquitetura bastante difundida de agentes cognitivos é a arquitetura BDI (abreviação de *beliefs*, *desires* e *intentions*) mostrada na Figura 1.

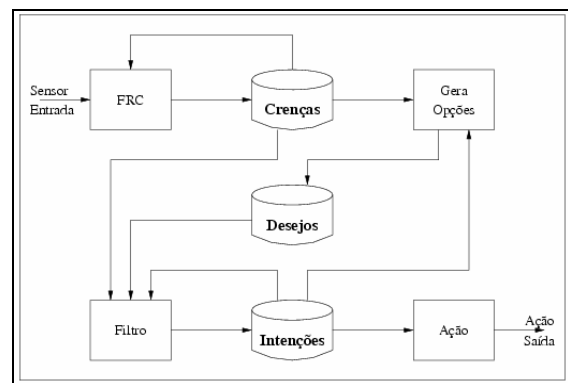


Figura 1 - Arquitetura BDI Genérica (adaptado de Wooldridge (1999)).

No contexto acima, a arquitetura BDI está esquematicamente definida da seguinte forma: As *crenças* representam a visão acerca do ambiente (seu estado) e também dos agentes envolvidos na sociedade; *Desejos* representam o estado em que o agente aspira atingir, ou seja, uma representação de estado no qual o agente quer que passe a ser verdadeiro no ambiente. Um desejo é composto por um conjunto de objetivos vinculados entre si; *Intenções* são seqüências de ações específicas que um agente se compromete a executar para atingir determinados objetivos. A *FRC* (função de revisão de crenças) tem a função de verificar as crenças anteriores para serem atualizadas resultando em um novo estado do ambiente a partir das percepções do agente. Atingindo um novo estado do ambiente é possível que novas alternativas de estados a serem atingidos fiquem disponíveis. O componente *Gera Opções* apresentado na Figura 1 tem por função verificar quais as novas alternativas de estado a serem atingidas. Este processo é realizado mediante as crenças do agente e nas intenções com que o agente está comprometido a realizar.

A atualização dos objetivos pode ser realizada de duas formas: através da observação do ambiente que podem determinar novos objetivos a serem alcançados e pela execução das intenções.

O *Filtro* é responsável por realizar a atualização das intenções mediante as crenças, os desejos e as intenções já existentes. Este processo é conhecido como deliberação [7].

Sistemas Multi-Agentes (SMA)

É um conceito de IAD adotado para o desenvolvimento de sistemas computacionais a partir dos agentes.

Conforme ilustrado anteriormente, os agentes são entidades de natureza autônoma que atuam no ambiente para modificá-lo de acordo com os seus objetivos a serem atingidos. Em algumas ocasiões surge à necessidade de tais agentes estabelecerem uma comunicação entre si para resolverem uma tarefa, o que torna necessário criar mecanismos de coordenação para que funcione corretamente. Sob este ponto de vista, os agentes são organizados em *sociedade*. Assim, “comportamentos

inteligentes podem emergir de uma população de numerosos agentes” [6].

Existem basicamente dois tipos de SMA [7] [8]:

- SMAC – Sistemas Multi-Agentes Cognitivos: sociedade composta de agentes cognitivos. Geralmente a sociedade é composta por poucos agentes devido a sua complexidade;
- SMAR – Sistemas Multi-Agentes Reativos: sociedade composta por agentes reativos. A sociedade é composta por muitos agentes.

Atualmente, tem-se adotado a fusão dos SMAC e SMAR para a resolução de tarefas denominado SMA híbrido, um modelo organizacional de sociedade que é composta por agentes cognitivos que compõem a classe coordenadora, e reativos que representa a classe operária. Os agentes coordenadores são responsáveis por delegar tarefas para os operários resolverem.

Metodologia

O processo de distribuição de materiais

O estudo de caso deste trabalho consiste de uma rede de distribuição de medicamentos de um almoxarifado gestor para as UBSs, conforme é ilustrado na Figura 2.

Regularmente e de forma sincronizada as UBSs são supridas com uma lista de medicamentos padrão mediante uma cota que varia entre as UBSs. O ciclo de distribuição é mensal.

Na véspera do término de cada ciclo, o usuário gestor responsável de cada UBS faz o processo de estimação de cota dos itens para o próximo mês. O recálculo das cotas de cada setor é enviado para o almoxarifado gestor, onde primeiramente são analisadas para posteriormente serem atendidas.

Cada UBS possui um sistema de controle de estoque de medicamentos para realizar a entrada / saída dos medicamentos.

O sistema de controle de estoque adotado neste trabalho que possui a arquitetura comentada na Figura 2 é o Sig2m. O Sig2m é um software para gerenciamento de almoxarifado, projetado para atender almoxarifados fisicamente distribuídos. Sua concepção foi desenvolvida a partir da realidade encontrada na estrutura organizacional dos centros de distribuição de medicamentos para postos de saúde, onde um almoxarifado central gerencia a compra,

armazenamento e distribuição de medicamentos e suprimentos para um conjunto de pequenos almoxarifados alocados em Unidades de Saúde espalhadas por uma ampla região, conforme. Atualmente o Sig2m é utilizado pelas prefeituras municipais de Campinas e Guarulhos.

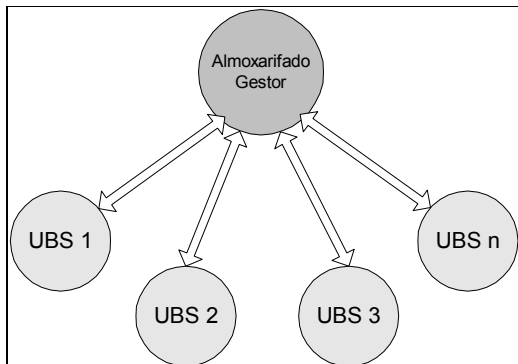


Figura 2 - Arquitetura da rede de Municipal de Saúde (UBSs e Almoxarifado Gestor).

Solução Proposta

Como proposta para a resolução da questão da má distribuição de medicamentos, foi montada a seguinte arquitetura: cada UBS contém uma sociedade de agentes híbrida formada por um *agente gestor* (agente cognitivo) modelado sob a arquitetura de estados mentais (BDI) e um *agente messageiro* (agente reativo) denominado *sociedade local*.

A função da sociedade local é auxiliar o usuário a monitorar o estado do estoque dos medicamentos. Esta função consiste em prever através de cálculos de gestão de estoque os medicamentos que estão excedentes (posteriormente serem doados para outras UBSs) e / ou que irão atingir a margem de risco de esgotamento (para posteriormente serem requisitados para as demais UBSs) antes do tempo de reposição e também calcular a cota ideal para o ciclo seguinte.

O usuário é informado sobre o estado do estoque através de mensagens enviadas pelo agente gestor intercedido pelo agente messageiro, responsável por prover o canal de comunicação. O agente messageiro obtém a percepção de uma mensagem enviada pelo

agente gestor e a encaminha ao sistema de controle de estoque, vide Figura 3.

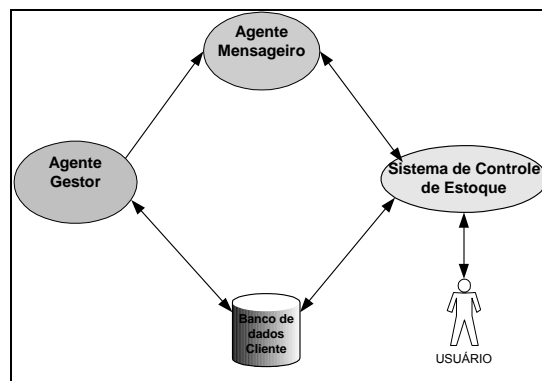


Figura 3 - Arquitetura da sociedade local.

O agente gestor faz parte de uma segunda sociedade, chamada de *sociedade remota*.

A sociedade remota consiste em uma sociedade de agentes gestores, sendo que cada agente gestor é representante de uma UBS. Tem por finalidade (neste contexto), gerar a comunicação entre os demais agentes cognitivos da sociedade para realizar o processo de requisição e / ou doação de medicamentos a partir da lista de medicamentos zerados e excedentes. Na véspera do fim de cada ciclo, o agente gestor envia a cota ideal de cada medicamento para o almoxarifado gestor.

Arquitetura das sociedades

Ambas as sociedades foram moldadas sob a arquitetura de comunicação orientada a agentes facilitadores, agente complexo que organiza o trabalho entre os agentes da sociedade. O facilitador é responsável por armazenar informações relevantes como a identidade, localização (endereço na rede) e serviços providos pelos agentes da sociedade [10].

Toda vez que um determinado agente entra na sociedade, este anuncia ao agente facilitador a sua entrada e publica as suas funcionalidades (passos 1 e 2 da Figura 4). Toda a comunicação entre os agentes é intermediada pelo agente facilitador.

Quando um agente A pretende se comunicar com o agente B, e não possui o seu endereço, este recorre ao agente facilitador (passo 3) para solicitar o endereço na rede do

agente B (passo 4) para então realizar a comunicação (passo 5).

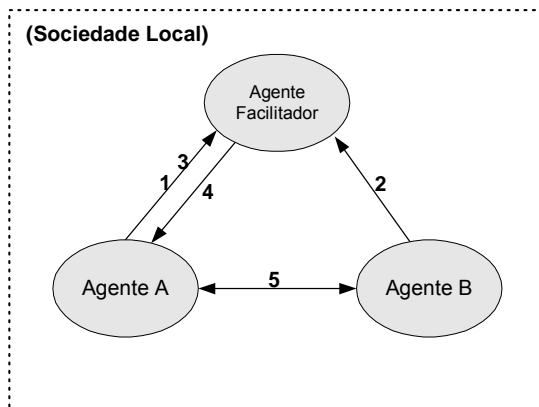


Figura 4 - Sociedade intermediada pelo agente facilitador.

Protocolo de comunicação entre agentes

A comunicação entre os agentes é assíncrona e é feita através de um protocolo de comunicação. O protocolo de comunicação adotado é o Knowledge Query and Manipulation Language (KQML), que dá suporte a alto nível de comunicação entre agentes para realizar a troca de conhecimento e informações [10] [11].

Funcionamento do sistema

Em cada ciclo, o agente gestor da UBS realiza cálculos de gestão de estoque para encontrar a cota ideal para o próximo ciclo. A partir de tais cálculos, é obtida a lista de itens que estão em estado de risco. A lista é constituída por itens que irão faltar e / ou sobrar em um período, bem como, a quantidade de dias prevista de duração do estoque.

Depois de realizados os cálculos de gestão, o agente gestor informa ao usuário sobre a situação do estoque.

Ao observar o estado do estoque, o usuário pode optar por doar e / ou requisitar medicamentos para outras UBSs, ou até mesmo para o almoxarifado gestor.

Caso a UBS necessite de medicamentos durante um ciclo, o usuário gera uma lista de requisição de itens a ser enviada para os demais UBSs. Então o agente gestor realiza a ação de encaminhar a lista de requisições para todos os outros demais agentes gestores

da sociedade remota, candidatos à doação, passos 1 e 2 da Figura 5.

O agente gestor doador realiza a percepção da chegada da lista de requisição de medicamentos, a armazena e realiza a ação de informar o usuário sobre a sua chegada.

Se a UBS candidata a doação possui itens excedentes, então o usuário poderá realizar a doação para o setor requisitante. A partir da lista de requisição, o usuário gera a sua proposta à doação. O agente gestor doador realiza a ação de enviar para o agente gestor requisitante a proposta de doação e espera a sua confirmação de aceitação e / ou rejeição (passo 3 da figura).

Ao receber a lista de proposta de doação, o agente gestor requisitante informa o usuário sobre a sua chegada. O usuário por sua vez, analisa a proposta de doação e confirma a aceitação e / ou rejeição (passo 4) que é novamente encaminhada para o respectivo setor doador para gerar a sua efetiva saída.

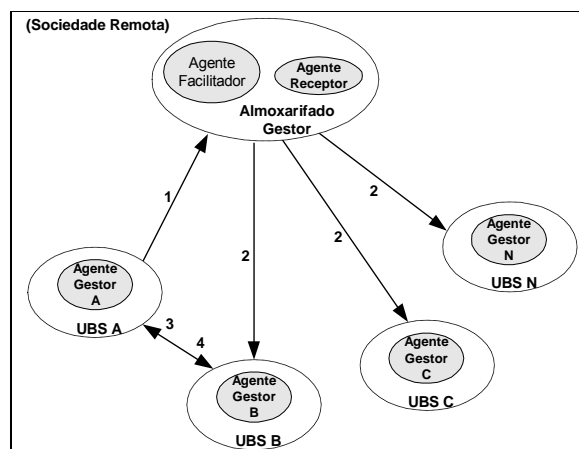


Figura 5 - Processo de requisição e doação de medicamentos entre UBSs.

Periodicamente, próximo ao termino de cada ciclo, cada agente gestor da sociedade envia para o almoxarifado gestor a proposta de cota ideal para o suprimento do ciclo seguinte com as cotas corrigidas.

Resultados

Apresentação do Estudo de Caso

Para a obtenção de resultados do estudo de caso, foi montada uma estrutura similar à realidade encontrada nos municípios. A estrutura utilizada na simulação possui 4 instâncias de UBS, sendo que cada UBS possui um sistema de controle de estoque e sua respectiva base de dados com informações reais com pelo menos 2 anos de uso. Também, foi criado um almoxarifado gestor delegado a realizar todo o processo de gestão de medicamentos em um determinado ciclo.

A duração de um ciclo, adotada no estudo de caso, é de um mês.

O objetivo do estudo de caso é verificar, de forma genérica, a convergência do estoque de medicamentos real para o estoque ideal (cota ideal) a partir das ações que os usuários da UBS realizam, orientadas pelos agentes gestores. Quanto mais próximo o estoque real se aproximar do ideal, significa que as irregularidades do estoque foram resolvidas através das ações realizadas pelos usuários auxiliadas pelo agente gestor, resultando em uma cota próxima da realidade para cada setor da rede de UBS.

Apresentação dos Resultados

Os resultados obtidos da simulação são apresentados nas tabelas **Erro! A origem da referência não foi encontrada.**1, Tabela 2, Tabela 3 e Tabela 4. Estas tabelas representam o consumo de um mesmo item em 4 UBSs distintas no período de um mês. Nas tabelas, a linha pontilhada representa o estoque ideal calculado pelo agente gestor e a linha contínua é o estoque real do setor.

Durante o ciclo, o agente gestor auxiliou o usuário no monitoramento e nas tomadas de decisões acerca do seu estoque. A estratégia adotada é aproximar o saldo real ao ideal através de ações realizadas pelo usuário. Em outras palavras, esta ação consiste em normalizar a situação dos itens que estão excedentes e zerados no estoque (cálculo realizado pelo agente gestor).

Os itens que iriam atingir a margem de risco foram requisitados aos demais setores da sociedade, através do agente gestor, como mostram os resultados da Tabela1 e Tabela 3, em que o determinado item estava em risco de faltar e então o usuário realizou a sua requisição para todos os setores por

intermédio do agente gestor. Os setores que possuíam tal item em demasia (como é o caso da UBS2 e UBS4) realizaram doação para os setores requisitantes. Na tabela 4 este comportamento pode ser notado como um ligeiro declive de seu saldo próximo a data de 22/03/2006. Já na tabela 2, a quantidade doada é pequena em relação ao seu estoque o que deixa quase que imperceptível no gráfico.

Com relação aos itens que possuíam quantidades desnecessárias às suas necessidades, os mesmos foram devolvidos para o almoxarifado gestor. Isto pode ser visualizado a partir da data 22/03/2006 nas tabelas Tabela 2 e Tabela 4.

Ao se aproximar o término do ciclo corrente, o cálculo da cota ideal dos itens de cada setor foi enviado para o almoxarifado gestor, para que no próximo ciclo tais setores recebam as cotas de medicamentos corrigidas.

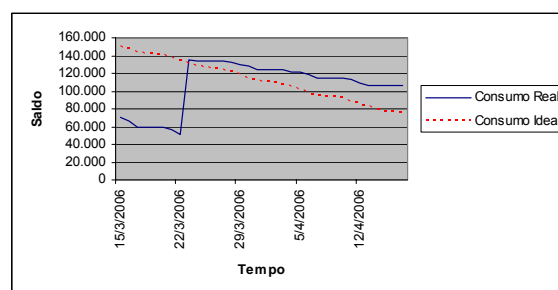


Tabela 1 - Consumo de medicamentos no período de um mês da UBS 1.

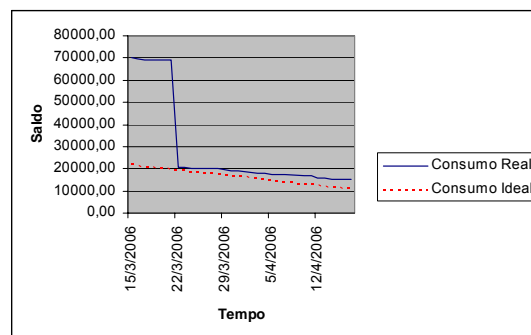


Tabela 2 - Consumo de medicamentos no período de um mês da UBS 2.

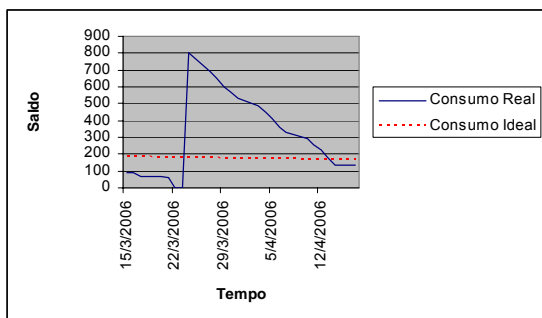


Tabela 3 - Consumo de medicamentos no período de um mês da UBS 3.

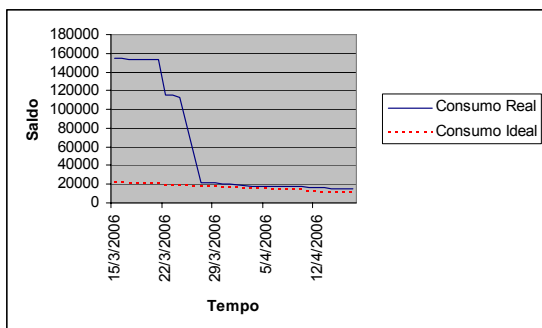


Tabela 4 - Consumo de medicamentos no período de um mês da UBS 4.

Discussão e Conclusões

Com base nos resultados obtidos, fica claro que o trabalho cooperativo do sistema multiagente cognitivo auxiliou os usuários a resolverem a questão da carência e o excesso de medicamentos nos setores, trazendo um certo equilíbrio ao estoque geral. Tal equilíbrio traz uma série de contribuições para a rede municipal da saúde, tais quais:

- A prevenção de gastos em demasia com medicamentos;
- A alta rotatividade de medicamentos, o que evita que os itens pereçam nos setores;
- Possibilidade de redução do estoque do almoxarifado centralizador.

Outro ponto bastante importante, é que foi comprovado que a carência de medicamentos nos setores não é por falta no estoque geral, mas sim, pelas falhas na distribuição dos medicamentos ocasionadas pelas solicitações equivocadas de cotas.

Diante destes fatos, pode-se dizer que é uma proposta atrativa o uso de sistemas Multi-Agentes cognitivos como ferramenta auxiliar para o monitoramento e gestão de estoque atuando na resolução da distribuição de medicamentos, as quais ainda são um dos principais problemas encontrados nas redes municipais de saúde.

Agradecimentos

Agradecemos a doutoranda Ana Carolina Gondim Inocencio pelo auxílio nas simulações realizadas. Ao doutorando Cláudio Roberto Araújo pelo apoio, esclarecimento de dúvidas e por ceder os sistemas de controle de estoque para a realização das simulações. E à prefeitura municipal de Campinas pelo fornecimento das informações utilizadas neste trabalho.

Referências

- [1] Finin, T., Weber, J., Wiederhold, G., Genesereth, M., Fritzson, F., McKay, D., McGuire, J., Pelavin, P., Shapiro, S. (1993), Specification of the KQML Agent-Communication Language, Technical Report EIT TR 92-04, Enterprise Integration Technologies, Palo Alto, CA, Updated in July.
- [2] Weiss, G. (1999), Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, p. 121 Ed. Gerhard Weiss.
- [3] Alvares, L. O., Sichman, J. S. (1997), Introdução aos Sistemas Multi-Agentes. In: Medeiros, Cláudia Maria Bauzer (Ed.). Jornada de Atualização em Informática (JAI'97). Brasília: UnB, cap. 1, p. 1–38.
- [4] Hermans, B.: Intelligent Software Agents on the Internet – An Inventory Currently Offered Functionality in the Information Society a Prediction of (near) Future Developments. Disponível em: http://www.firstmonday.dk/issues/issue2_3/ch_123/index.html. Acessado em: 07/04/2006.
- [5] Bordini, R. H., Vieira, R., Moreira, Á. F. (2001), Fundamentos de Sistemas Multi-Agentes. In: Congresso da Sociedade Brasileira de Computação (SBC2001), XX Jornada de Atualização em Informática (JAI), 21. Anais. Fortaleza-CE, Brasil: Sociedade Brasileira de Computação. p 3–41.
- [6] Halfpap, D. M., Belli, M. J. (1997), O Papel de Sistemas Inteligentes para o

desenvolvimento de Organizações Virtuais no ambiente, Internet, Universidade Federal de Santa Catarina. [www.ufsc.br]. Acessado em 10/01/2006.

[7] Hübner, J. F., Bordini, R. H., Vieira, R. Introdução ao desenvolvimento de Sistemas Multi-Agentes com Jason. Departamento de Sistemas e Computação, Universidade Regional de Blumenau (FURB). p. 1-12.

[8] Giraffa, L. M. M. (1999), Uma arquitetura de tutor utilizando estados mentais, Tese de Doutorado, UFRGS - Universidade Federal do Rio Grande do Sul, Instituto de Informática

Contato

Ekler Paulino de Mattos – Bacharel em Ciência da Computação, mestrando pela CPG-FEEC - Programa de Pós-Graduação da Faculdade de Engenharia Elétrica e Computação - Unicamp – Universidade Estadual de Campinas.

Leonardo de Souza Mendes – Professor Doutor associado da Faculdade de Engenharia Elétrica e Computação - Unicamp – Universidade Estadual de Campinas.

Mauricio Bottoli – Engenheiro Eletricista, doutorando pela CPG-FEEC - Programa de Pós-Graduação da Faculdade de Engenharia Elétrica e Computação - Unicamp – Universidade Estadual de Campinas.

FEEC - UNICAMP
Cidade Universitária Zeferino Vaz
Av. Albert Einstein - 400 - 13083-852
CEP 6101 - 13083-970 - Campinas - SP
Tel: (19) 3788-3829

Email: ekler_mattos@yahoo.com.br
lmendes@decom.fee.unicamp.br
mauricio.bottoli@gmail.com

Programa de Pós-Graduação em Computação, Porto Alegre, 43-59p mai.

[9] Sistema de Gestão de Materiais e Medicamentos. Disponível em: http://www.igniscom.com.br/prod_publico/sig2m.php. Acessado em: 10/06/2006.

[10] Hübner, J. F., Sichman, J. S. (2003), Saci Programming Guide, USP - Universidade de São Paulo, São Paulo, jul.

[11] Genesereth, M.R., Ketchpel, S. P., (1994), Software Agents. Communications of the ACM. Vol. 37 (7). July 48-53.